CALIBRATION AND DATA REDUCTION

OF AN UV SPECTROPHOTOPOLARIMETER

A THESIS

Presented to

The Faculty of the Graduate Division

by

Ellis Bayley Hodgdon

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Physics

Georgia Institute of Technology

October, 1966

CALIBRATION AND DATA REDUCTION

OF AN UV SPECTROPHOTOPOLARIMETER

Approved:

Chairman

Date approved by Chairman: 10/28/66

ACKNOWLEDGMENTS

The author wishes to thank Dr. Howard Edwards for his valuable assistance and suggestions. Drs. C. D. Cooper and E. Rhodes were also responsible for many comments and suggestions. The author wishes to express his appreciation to his colleagues at the Space Sciences Branch at Georgia Institute of Technology, Mr. Milford Brown of the Balloon Branch at Holloman Air Force Base, New Mexico, and Mr. Cal Skelton of Telecomputing Services, Inc. who helped make the flight possible. Mr. Bill Simpson has helped in the preparation of the diagrams. Mrs. Maria Elena Austin is responsible for the neat execution of the manuscript.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

SUMMARY

A spectrophotopolarimeter, which is designed to measure the intensity and polarization of skylight radiation at altitudes of greater than 100,000 feet, produces a large volume of data and must be suitably calibrated prior to any measurements. The instrument uses imperfect optical elements, and the effect of these optical components is determined by the calibration procedure. Calibration is simple to perform, but, like the data reduction which follows, requires the use of digital computers. The data reduction involves manipulating the data so that they are in a form to be reduced, and then performing the actual computations.

The spectrophotopolarimeter is a combination of two instruments: one, a spectrometer which uses photomultiplier tubes to measure the wave length range of 2000 Å to 4000 Å with 16 Å resolution, and the other, a polarimeter which uses a rotating retardation plate followed optically by a linear polarizer. The output of the photomultiplier tubes is a composite AC signal which has a DC component and harmonics of twice and four times the rotational frequency of the retardation plate. By analyzing these components, the polarization and the intensity can be determined. The output of the photomultiplier tubes is digitized at a high rate so that the harmonics can be accurately determined.

Some actual data have been obtained, but their presentation only demonstrates that the calibration procedure and the data reduction procedure are performing satisfactorily. No attempt has been made to reduce the large volume of data that is produced, but only enough to show that the data reduction chain is feasible.

CHAPTER I

INTRODUCTION

A balloon-borne ultraviolet (2000 to 4000 Å) spectrophotopolarimeter, which is designed to measure the intensity and polarization of the natural sky backgrounds at an altitude of 120,000 feet, generates a large volume of data during a typical flight of approximately eight hours. The spectrometer portion of the system is basically an Ebert-Fastie arrangement with a 50 cm. focal length mirror and theoretical f/5 optics.[1,2] The polarization measurement portion is essentially a Sekera polarimeter,[3-5] which consists of a rotating retardation plate followed optically by a linear polarizer. The light passing through this linear polarizer is measured by a photomultiplier tube. The intensity and polarization parameters can be determined by analysis of the AC components of the signal from the PM tube.

The actual hardware that was flown is described in detail elsewhere,[6] but a brief description will be repeated so that the flow of information can be followed from the light that strikes the entrance slit of the spectrometer to the actual numbers for intensity and polarization data. Although the mathematics is not complicated, the large amount of data that is produced requires the use of electronic computing equipment. This thesis will give a detailed description of the data handling techniques that have been used on previous flights, including the steps through which the data must pass before the results can be obtained. The instrument and the calibration procedure are unique in that no perfect optical components are required anywhere, e.g., an imperfect linear polarizer is one that permits some of the unwanted orthogonal component of light to pass through, and an imperfect retardation plate is one which exhibits some additional linear polarization besides retarding the two orthogonal components.

Although some actual data of skylight intensity are presented, the main purpose of this thesis is to show the process by which these data were obtained and the process by which the instrument was calibrated in order to obtain these data. Consideration is always given to the effect of imperfect optical elements and the amount of imperfectness is determined by the calibration procedure.

A general idea of the flight package can be obtained from Figures 1 and 2. These pictures show the spectrometer, yoke and gondola just prior to the 1 July 1965 launch, and important features are as follows.

In Figure 1: The detector head contains all the components necessary to measure the amount of light passing through the exit slit of the spectrometer. The quarter-wave plate and its drive mechanism provides a method of determining the polarization of the incident radiation. The sunshade limits the field of view of the spectrometer and eliminates some reflected light, and the guard cell sensors activate an auxiliary "semi-shutter" which reduces the light entering the spectrometer by approximately two orders of magnitude. The coarse PC (pointing control) sensors provide an error signal to the PC amplifier when the spectrometer is pointing considerably away from the sun, while the fine eye block provides the error signals for fine positioning of the spectrometer system reference axis, which the pointing control attempts to maintain pointed at the sun at all times. The elevation servo provides the torque necessary to point the

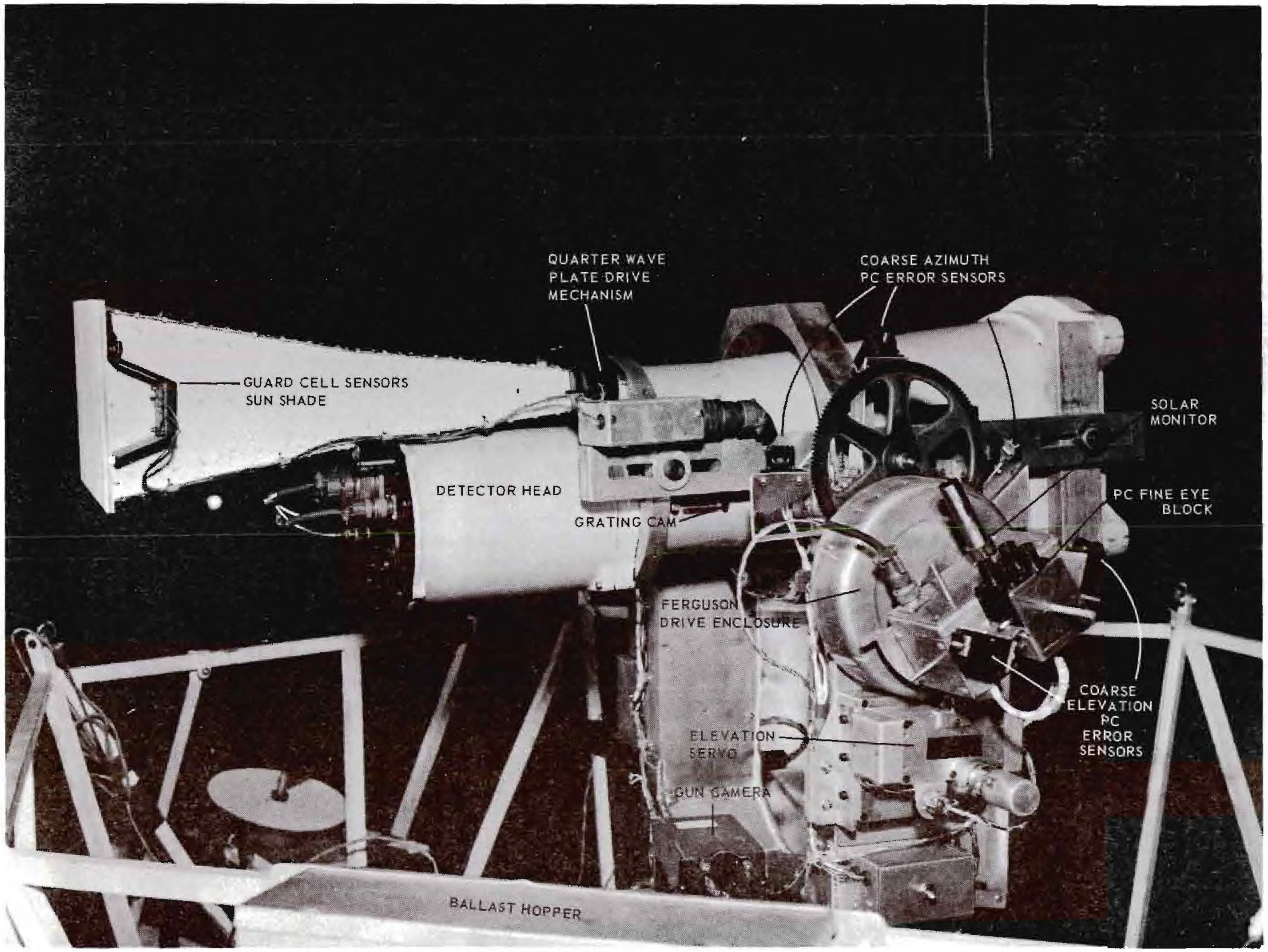

Figure 1. Spectrometer Showing Major Components.

spectrometer system reference axis at the sun. The Ferguson stepping drive imparts an intermittent motion to the spectrometer. Under the influence of this drive the spectrometer is stationary for 30 seconds and moves 15 degrees with respect to the spectrometer system reference axis in the next 30 seconds. The solar monitor provides a means of determining an "on-sun" condition and provides the reference axis from which the spectrometer steps.

In Figure 2: The ballast hoppers, supplied by Air Force Cambridge Research Laboratories (AFCRL), are one means of controlling the ascent rate of the balloon. The yoke assembly holds the spectrometer while the spider supports the yoke and provides some shock protection. The control panel serves as an electrical junction box and provides means of controlling some functions of the flight package and monitoring some of the systems. The gun camera takes a picture of the spectrometer every two minutes through a convex rear view truck mirror to serve as a backup for the pointing control. (If the pointing control were to fail and lock onto something other than the sun, the position of the spectrometer could be determined from the film.) The flight control package, also provided by AFCRL, provides three switch closures by radio command from the ground: drop ballast, telemetry on-off, spectrometer package on-off. The vehicle from which the package was launched can be seen in the background, and some of the equipment used during the pre-flight checkout can be seen on the ground beside the package.

Three actual flights have been made with the basic package, but the first two fell somewhat short of being a total success. The first flight, on 26 August 1964, was aborted shortly after reaching float altitude because of a premature failure of the electronics batteries and stoppage of

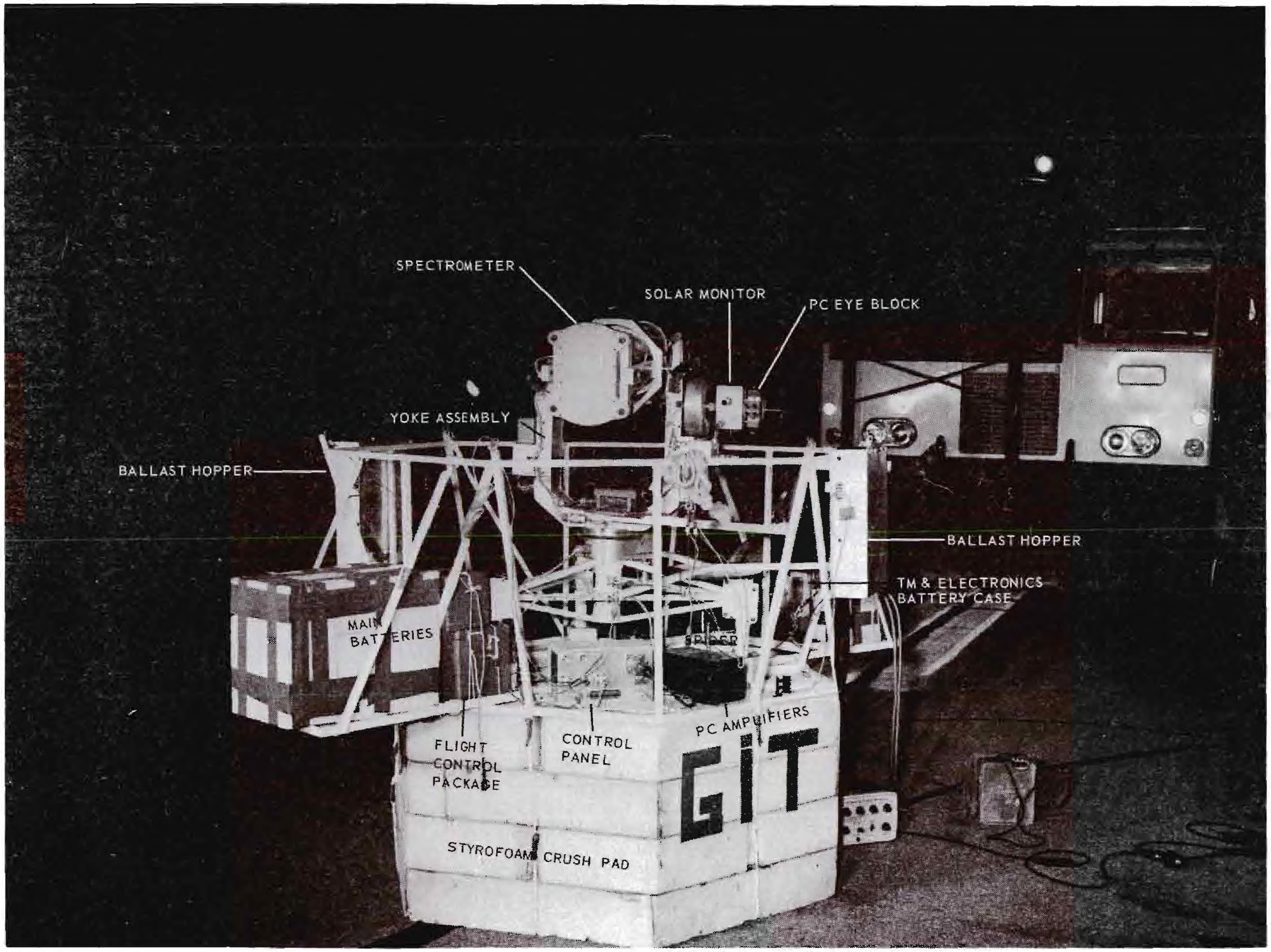

Figure 2. Gondola Showing Major Components.

the quarter-wave plate drive. The pointing control also failed to perform properly. The instrument suffered major damage on impact and the optics of the spectrometer were fouled by transportation over unfavorable terrain back to the launch area.

The second flight, on 1 July 1965, was a trifle more successful. A slight mixup of instructions and drift of the telemetry caused the data not to be received satisfactorily until after the balloon had reached 100,000 feet. A more serious problem developed since the FAA requires that visibility flags be flown every fifty feet when the gondola and the balloon are separated by a long load line. The device which was to deploy these flags singly as the gondola dropped beneath the balloon via the let down reel malfunctioned and deployed all the flags before the gondola had dropped sufficiently. As a result, the flag line became entangled in the spectrometer yoke and prevented the pointing control from operating properly.

From a portion of the data, it was discovered that the spectrometer had randomly found the sun for approximately one hour and forty-five minutes. This time was accumulated mainly after reaching float altitude when the gyrations of the gondola were such as to bring the pointing control and solar monitor around to the sun. It is data recorded at these "on-sun" times which are presented later.

The third flight on 12 July 1966, seemed to be more successful, although sufficient data have not been reduced at the present to ascertain porper operation of all systems. It is known that a wire broke on the elevation scan potentiometer, which gives an indication of the relative angle between the spectrometer and the spectrometer system reference axis, so that the gun camera data will have to be used to determine the scatter angle of the radiation being measured, but this is not believed to be a serious problem. With each flight the instrumentation package is improved and modified, although the changes are never major and the basic design remains intact.

The brief description of the instrumentation package follows, the flow of information being emphasized. The data reduction techniques and procedures are described in detail, and some limited results from the 1 July 1965 flight are also given.

CHAPTER II

INSTRUMENTATION PACKAGE

Data from several sources must be combined in order to obtain the final results of intensity and state of polarization of the skylight radiation at a given scatter angle from the sun and at a given wave length. To obtain the intensity and polarization of the incident radiation, the signals from the two photomultiplier tubes and the AC generator of the polarimeter must be utilized, while a grating switch signal and the output from the elevation scan potentiometer give the wave length and scatter angle respectively. All the data from the balloon package are telemetered in real time to the ground and recorded on analog tape recorders.

A mathematical description of a polarized light optical system is facilitated by the use of Stokes vectors and Mueller calculus (see Appendix A). A Stokes vector is a four element column vector which completely describes the intensity and state of polarization of radiation, while Mueller calculus uses four by four matrices to describe the effect that optical components have on light. The Stokes vector of the transmitted radiation can be obtained by operating on the Stokes vector of the incident radiation with the proper Mueller matrix for an optical system.

Photomultiplier Tube Signals

The basic optical design of the complete spectrophotopolarimeter is shown in Figure 3. Light entering the entrance slit S_1 strikes the spherical mirror M and is collimated toward the grating G where it is refracted according to the wave length relation for Ebert geometry:

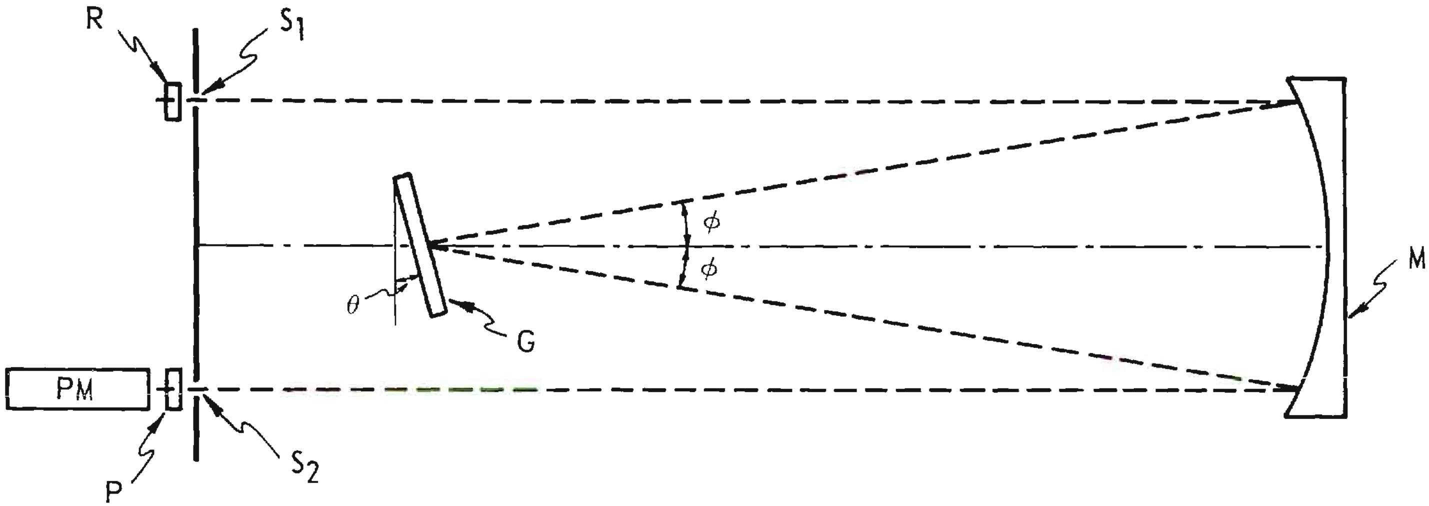

Figure 3. Optical Arrangement of the Ebert-Fastie Spectrometer-Polarimeter.

$$n\lambda = 2\,d \cos \Phi \cos \theta \quad , \tag{1}$$

where n is the spectral order, λ is the wave length that appears at the exit slit S_2, Φ is the angle between the axis of symmetry and the light incident upon the grating, and θ is the angle of rotation between the grating normal and the axis of the instrument. The diffracted light of wave length λ, after leaving the grating, strikes the mirror and is refocused on the exit slit, through which it passes and is detected and measured by the photomultiplier tube PM. The retardation plate R and the linear polarizer P are components of the polarimeter and will be described later.

The grating is driven by a cam and cam follower mechanical arrangement. The cam is driven by a governed 3 RPM DC motor and is cut so that a linear relation exists between the rotational angle θ and the wave length λ which appears at the exit slit. The cam is cut by the manufacturer of the basic Ebert-Fastie spectrometer for a particular grating ruling (in this case, 2160 grooves/mm.) and the wave length range to be covered.

Two photomultiplier tubes are used to cover the 2000 to 4000 Å range. One is a "solar blind" tube used primarily below 3100 Å while the other is used in conjunction with a Corning #7-54 filter for a 2800 to 4000 Å band pass. Figure 5 shows a calibration parameter, C, which is a measure of the spectral sensitivity of the entire system, including the spectral response of the photomultiplier tubes. Each photomultiplier tube signal is amplified by a high impedance nonlinear amplifier which has a dynamic range of approximately four log cycles. The output of these amplifiers is 0 to 5 volts, which is the input requirements for the telemetry system.

The photomultiplier tubes, amplifiers, and the associated subsystems are contained in a pressurized detector head mounted on the slit end of of the spectrometer (see Figure 1). The detector head contains a DC-DC converter which supplies high voltage (up to 3000 volts) for the photomultiplier tubes and a regulated center-tapped power supply which supplies power for the amplifiers and the DC-DC converter. Since the nonlinear characteristic of the PM tube amplifiers is temperature sensitive, the amplifiers are contained in a temperature controlled (+ 85 °F ± 3 °F) oven for stability. A small calibration light source, which provides an input light to the PM tubes during the calibration cycle to obtain a nominal calibration point, and its current regulated power supply are also contained in the detector head as well as monitoring and calibration circuitry. The airtight construction of the detector head is such that it will maintain a nominal one atmosphere pressure even at float altitude so that the high voltage present will not arc.

Polarimeter Design and AC Generator Signal

Mathematical Description of Polarimeter

Since polarization is present in most spectrometers which employ a grating as a diffracting element, some method of analyzing the polarization of the incident light was decided upon over elimination of the polarization sensitivity of the spectrometer. A Sekera-type polarimeter was constructed and attached to the instrument, and the mathematics evolving from the use of retardation plates and linear polarizers such as in this polarimeter need some explanation. This polarimeter requires a linear polarizer as part of the optical train, but since no polarizer could be found that would transmit sufficiently below 2300 Å, the spectrometer grating itself, which is an imperfect polarizer, was used for the linear polarizer for the 2000 to 3200 Å tube.

By placing an imperfect retardation plate of retardance δ, not necessarily precisely $\lambda/4$, in front of a fixed imperfect linear polarizer (in this case, the entire spectrometer system and the polarizer), the intensity and the other Stokes parameters can be found from a partial inverse solution of

$$[A] = [P_o] [R_\beta] [I] , \tag{2}$$

where $[A]$ is the Stokes vector of the light leaving the linear polarizer, $[I]$ is the Stokes vector of the incident radiation, and $[P_o]$ and $[R_\beta]$ are defined by equations (A-17) and (A-28), respectively.

The photomultiplier signal is proportional to the intensity of light striking the photocathode, which can be placed in the following form:

$$I_A = a_1 + a_2 \sin 2\beta + a_3 \cos 2\beta + a_4 \sin 4\beta + a_5 \cos 4\beta \tag{3}$$

where the a's are Fourier coefficients that have a definite form which is shown in Appendix B and where β is the angle the fast axis of the retarder makes with a reference axis. After the components a_1, a_2, a_3, a_4, and a_5 are determined by some method of harmonic analysis, the unknown quantities, the modified Stokes parameters of the incident light, I, P, V, and Φ, are given by

$$I = \frac{1}{C\left(1 + r_p^2\right)\left(1 + r_r^2\right)} \left[a_1 - \frac{\left(1 + r_r^2 + 2r_r \cos \delta\right)}{\left(1 + r_r^2 - 2r_r \cos \delta\right)} a_5 \right] \tag{4}$$

$$P = \frac{2\left(a_4^2 + a_5^2\right)^{\frac{1}{2}}}{CI\left(1 - r_p^2\right)\left(1 + r_r^2 - 2r_r \cos \delta\right)} \tag{5}$$

$$V = \frac{a_2 - CI\left(1 + r_p^{\,2}\right)\left(1 - r_r^{\,2}\right) P \sin 2\phi}{2CI\, r_r\left(1 - r_p^{\,2} \sin \delta\right)} \tag{6}$$

$$\phi = \tfrac{1}{2} \tan^{-1}\left(\frac{a_4}{a_5}\right) \tag{7}$$

The parameters C, r_r, r_p, and δ are characteristics of the system and are determined by the calibration procedure. Equations (4) through (7) will therefore determine the modified Stokes parameters of the incident light from the Fourier components of the exit intensity as a function of β.

The parameter δ is the retardance of the phase plate, r_p is the ratio of the minor axis transmission to the major axis transmission for the linear polarizer (ideally, $r_p = 0$), r_r is the ratio of the slow axis transmission to the fast axis transmission for the retarder (ideally, $r_r = 1.0$), and C is the spectral sensitivity of the system.

Initially, the spectrometer was assumed to be only a linear polarizer. However, a comparison of equations (A-17) and (A-25) reveals that the two Mueller matrices for an imperfect linear polarizer and an imperfect retardation plate are similar. Since the intensity is the only quantity measured by the photomultiplier tubes and since the spectrometer is last in the optical train, only the first row of the Mueller matrix which represents the spectrometer is needed. The first row of both the linear polarizer matrix and the retardation plate matrix are identical in form so the spectrometer could be considered to be either an imperfect linear polarizer or an imperfect retardation plate and the results would be the same. This means that the spectrometer could have elliptically polarizing elements, e.g., metallic reflectors, and the results to be derived would be the same as the case of the spectrometer exhibiting only linear polarization.

Effect of Imperfect Optical Elements

Since it was known beforehand that the optical components used were not perfect, it was necessary to determine the effect of such imperfect elements in the optical train. The main component to consider is the linear polarizer, since it is not nearly as perfect as the Glan-Thompson prism used by Sekera. Also, since the spectrometer covered a wide wave length region, a dependence on wave length of the various physical parameters may be considered an imperfection and must be considered. The relationships expressed in equations (9) through (12) form the basis for determining the effect of imperfect optical elements on the determination of the Stokes parameters.

It can be seen that if r_p is known with a fair degree of accuracy and is not unity, then all the polarization parameters can be uniquely determined. If r_p were unity, i.e. no polarizer, then although P, V, and Φ could not be determined, I could be measured if I were the only quantity of interest, and the requirement for a polarimeter would vanish. While P is proportional to $1/(1 - r_p^2)$, V is essentially proportional to $1/(1 - r_p^2)^2$, so any error in the determination of r_p will have some effect on P, but it would have a greater effect upon the measurement of V.

Other imperfect optical elements will have an effect on the determination of the Stokes vector of the incident light. Investigating the effect of the retardance of the phase plate on the Stokes parameters, we find that if $\delta = 0$ or 180°, then although V could not be determined, I, P, and Φ could. As expected, $\delta = 90^\circ$ leads to no difficulty in any determination. The other extreme on the phase plate would be if $r_r = 0$, which would correspond to the phase plate being a perfect linear polarizer. If this were the case, then again V could not be determined, but I, P, and ϕ could be.

In general, the requirements placed on the optical elements in the train would be that in the wave length region of interest $r_p < 1$, $r_r > 0$, $0^o < \delta < 180^o$, and obviously, $C > 0$. If these conditions are satisfied, then the four modified Stokes parameters (and consequently, the Stokes parameters themselves) of the incident light can be uniquely determined. However, even with these conditions fulfilled and all optical components properly calibrated, thermionic emission in the photomultiplier tube and other sources of noise in the system may seriously affect the measurements if, for example, r_p were close to unity. Therefore, the practical limit on the magnitudes of r_p, r_r, δ, and C which can be tolerated is set by experimental conditions.

It is known that C shows a strong wave length dependence since it includes, among other things, the spectral response of the photomultiplier tube, δ will exhibit a medium wave length dependence, and r_r and r_p will have some slight wave length dependence. The spectrophotopolarimeter would have to be calibrated at as many wave lengths as possible in order to observe these wave length dependencies and also to judge a reasonable confidence level which can be placedon the results.

Polarimeter Construction

The polarimeter which was actually used consisted of two separate systems, one for each photomultiplier tube, each using nominally one-half of the exit and entrance slits. The two retardation plates (R in Figure 3) were rotated at 20 RPS by a governed 4800 RPM DC motor through a 4:1 gear reduction. An AC generator was driven through a 2:1 gear reduction from the same motor. In principle, if the peak amplitude of the generator output were known, then the arc sine of the ratio of the amplitude at a given time to the peak amplitudes would give the angle of the generator shaft with respect to some reference axis at that time. This angle would then be equal within some constant phase angle to twice the angle of the retardation plate. It can be seen from equation (3) that multiples of the angle of the retardation plate are needed to perform the required harmonic analysis. The output of the AC generator is then fed into the telemetry system.

The linear polarizer (P in Figure 3) for the longer wave length tube is a Polacoat coated quartz disc mounted inside the detector head and aligned so that its axis is along the axis of maximum polarization of the spectrometer. For the shorter wave length tube, since no suitable polarizer could be obtained that would transmit below 2300 Å and since the instrument is a fair polarizer in the 2000 Å to 3000 Å, the instrument itself is used as the linear polarizer and no polarizer is used in the detector head.

Support Information

Two additional pieces of information which relay wave length and scatter angle are needed to complete the minimum information necessary for data reduction. Other information such as temperature, pointing control error signals, and battery voltages are transmitted, although they are not directly involved in the data reduction. Information of a slowly varying nature such as temperature, scatter angle, battery voltages, and error signals are combined by a commutator into one telemetry channel. The wave length data are transmitted on a separate channel.

A ratchet type cam is physically attached to the cam which rotates the grating. This ratchet cam is unevenly cut so that a microswitch which is activated by the cam is turned on and off at irregular times during one rotation of the cam, but the cycle is repeatable between rotations. For the 1 July 1965 flight, the points at which the grating switch was turned off corresponded to 2086, 2595, 3264, 3774, and 3900 Å. Thus it is possible to find the minimum time interval (and therefore the minimum wave length interval—in this case 126 Å) between consecutive pulses and set the last time of the interval in this case to 3900 Å, and from that point determine the wave length at any time during the forward sweep of the grating. The assumption is made that the grating cam is cut so that a linear relationship exists between the transmitted wave length and the angle of rotation, and a secondary assumption is made that the governed DC motor which drives the grating cam rotates at a constant speed for one rotation. It was felt that a switch closure and opening provided greater resolution than other means such as a potentiometer.

A precision potentiometer is mounted so that its resistance is a measure of the angle between the spectrometer system reference axis, which is, in operation, the line from the instrument to the sun, and the spectrometer axis. A suitable voltage divider provides an output between 1.0 and 5.0 volts (input voltage requirements for the commutator). This potentiometer is calibrated upon final assembly of the spectrometer since the sprocketed drive chain is disconnected when the instrument is shipped.

The other data that are transmitted via the commutator are adjusted by suitable means so that the voltage range is 1.0 and 5.0 volts. These data are commutated along with the scatter angle data (elevation scan potentiometer) onto one telemetry channel. Thus telemetry requirements are for four channels for continuous data—two PM tubes, the AC generator, and grating switch—and one channel for commutated data.

CHAPTER III

OPTICAL CALIBRATION

Objective

Since quantitative data were required to measure the skylight intensity, the optical system of the instrument had to be calibrated in terms of watts per cm^2 volt out. One method for such a calibration is to measure the characteristics of each component in the optical train and then figure the combined characteristic from the individual ones. However, for expediency and simplicity, a method was devised that permitted the optical calibration of the spectrophotopolarimeter in a maximum of two steps at each wave length of interest. It is possible with this method actually to calibrate the system in one step, if one knows the quantitative amount of light passing through an auxiliary polarizer. There is one serious disadvantage to this method, and that is the quantity of data that is required for calibration. Hand reduction of these data is practically impossible from the time standpoint, and therefore, the computational procedures have to be programmed for the computer. Another problem arises, however, namely the generation of the data in a form that is usuable by the computer, but this problem has been essentially solved, and a means has been obtained of reducing the large volume of calibration data. An analog-to-digital converter (ADC) at Telecomputing Services, Inc. uses the analog tape which is generated by the calibration procedures and produces a digital tape in IBM 7094 format which can be used on the Georgia Tech computer. Thus, the entire problem of data handling is essentially alleviated by computer and computing machinery.

The basic problem of calibration can be broken into three categories: (1) the calibration of the quarter-wave plate as to retardance, transmission, and linear polarization, (2) the calibration of the spectrometer for transmission and linear polarization, and (3) the calibration of the photomultiplier tubes for spectral response. The third problem is best solved by means of an NBS Standard of Spectral Irradiance[7] for which NBS has calibrated the light output versus wave length. It is assumed, however, that transmission losses in the optical train can be accounted for by modification of the spectral response of the photomultiplier tubes. Thus, the spectral response of the whole system which the calibration procedure produces is quite likely to differ from the spectral response furnished by the manufacturer.

For all the different calibrations except the quantitative value of light input versus voltage output, the procedure is fairly simple. A suitable light source (a GE quartz-iodine lamp was used) is placed behind a linear polarizer that can be rotated in fixed increments. The light from this polarizer is then allowed to fall on the entrance slit of the spectrophotopolarimeter. The output is then recorded by some means, either through telemetry onto magnetic tape recorders or by oscillographs from which the data is obtained by ADC or hand reduction, respectively. The angle of polarized light is stepped in its fixed increments until an angle of at least 180° has been passed. For a given wave length, the output is first broken down into components of the frequency of the rotating quarter-wave plate frequency and its harmonics, and then these components undergo a Fourier analysis with the angle of incident polarization as a basis. The system has the distinctly advantageous property that no optical component (quarter-wave plate, linear polarizers, etc.) has to be perfect because the system allows for imperfections in these devices and actually computes the amount of imperfection.

The calibration for the quantitative value requires only the NBS standard of spectral irradiance which is placed before the entrance slit. The output of the photomultiplier tube is analyzed into components of the rotating quarter-wave plate frequency, and the value of the transmission coefficient, C, for the entire system is calculated.

Specific Procedure

An accessory was constructed that was capable of rotating a linear polarizer in convenient fixed increments of 11.25 degrees. The polarizer was made from a sheet of Polaroid HNB. For the first part of the calibration procedure, this accessory was placed between the light source, a GE quartz-iodine lamp, and the entrance slit of the spectrophotopolarimeter. The polarizer was then rotated by manually pushing a switch at a time which corresponded to the return sweep of the grating. Thus, the polarizer was rotated every twenty seconds, which means that throughout the calibration run, for every sweep of the grating through the wave length region, there was a different angle of polarization of incident light.

For the second part of the calibration procedure, the NBS standard of spectral irradiance was placed a known distance (61.2 cm) from the entrance aperature of the spectrophotopolarimeter. Although this is not the distance at which the lamp was calibrated, it is reported that the inverse square law is valid at distances over 43 cms. Several wave length scans were made with the standard lamp as the light source to serve for calibration purposes.

The calibration procedure is definitely computer oriented. A large volume of data is accumulated, and some of the calculations, although they could be calculated using a desk calculator, are best left for the digital computer. For the utilization of the digital computer, a method of multiple linear regression was used which minimizes a function, G, of the form:

$$G = \sum_{i=1}^{N} \left[y_i - a_1 - a_2 x_{1i} - a_3 x_{2i} - a_4 x_{3i} - a_5 x_{4i} \right]^2 \qquad (8)$$

by adjusting the a's where, in this case, y_i is the photomultiplier tube output corrected for dark current, $x_{1i} = \sin 2\beta_i$, $x_{2i} = \cos 2\beta_i$, $x_{3i} = \sin 4\beta_i$, and $x_{4i} = \cos 4\beta_i$, where β_i corresponds to the β in equation 3 and the i^{th} value of y.

During a normal computation, a minimum of 25 samples are used in the determination of the a's; the number of samples is a function of the wave length interval for which the computation is made. At the present limit of analog-to-digital conversion, 500 samples per second per channel, approximately five samples are obtained as the grating sweeps through 1 Å. Since the resolution of the spectrometer is approximately 16 Å, the intensity is assumed to remain constant during a sweep of 5 Å (25 samples), and no correction is attempted for an intensity change during the sampling time.

The Fourier coefficients [the a's in equation (8)] are not obtained via a sine or cosine integration, but rather are obtained by a curve fitting process. A test was made to determine if higher order components of β were present, and the higher order coefficients were reduced in magnitude by a factor of 50 or higher. Also, a frequency spectrum analysis was performed on the photomultiplier signal, and similar results were obtained, i.e., only a DC, 2β, and 4β components were measurably present.

The results derived in Appendix B show that the calibration parameters— r_r, r_p, C, and δ (equations (4) to (7))—can be determined from

$$r_r = \sqrt{\frac{1 - k}{1 + k}} \tag{9}$$

$$\cos\delta = \frac{d_2 - d_6}{d_2 + d_6}\left(\frac{1 + r_r^2}{2r_r}\right) \tag{10}$$

$$r_p = \sqrt{\frac{kd_1 - d_5}{kd_1 + d_5}} \tag{11}$$

$$CI = \frac{d_1(1 + k)}{2(1 + r_p^2)} \quad , \tag{12}$$

where

$$k = \sqrt{\frac{d_4 d_5}{d_1(d_2 + d_6)}} \quad , \tag{13}$$

and where the d's are defined in Appendix B.

If the total intensity of light falling on the entrance aperture after passing through the polarizer is known, then C can be determined from equation (12). However, if the quantitative value of light passing through the polarizer is not known and a separate determination is needed to determine C, then after performing the required harmonic analysis on the PM tube signal, C can be found from

$$C = \frac{1}{I(1 + r_p^2)(1 + r_r^2)}\left[a_1 - \frac{1 + r_r^2 + 2r_r\cos\delta}{1 + r_r^2 - 2r_r\cos\delta}\,a_5\right] \tag{14}$$

If the source is completely unpolarized, then P = 0, and consequently, $a_5 \approx 0$. With the lamp used, a_5 was approximately three orders of magnitude small than a_1.

Calibration is complete after the determination of C, whether it comes from equation (12) or (14). No "perfect" optical components, except perhaps for an unpolarized light source, are used, and the calibration could be performed in just one operation. Individual components are not calibrated as such, rather the entire spectrophotopolarimeter is calibrated as a basic unit.

The basic computer program for the calibration procedure is shown in Appendix C while the flow diagram for this program is shown in Figure 4. The calibration data undergoes the same preparation as the regular data, i.e., the input tape has all data digitized, i.e., two photomultiplier tubes and the output of the AC generator, and the wave length calculated for the forward sweep of the grating. No wave length computation is attempted during the return sweep of the grating.

The results of the calibration program are used as input to a multiple linear regression library program which provides a mathematical approximation to the actual wave length dependence of the calibration parameters by fitting the calibration parameter to a specified function of wave length. This regression merely smoothes the data, removes bad calibration points and provides a convenient method of evaluating the calibration parameters. The coefficients which are thus determined are used in the main data reduction program. The calibration curves for C, r_r, r_p, and δ for the 1 July 1965 flight are reproduced in Figures 5, 6, 7, and 8 respectively.

The various coefficients in equations (4) through (7) [for example,

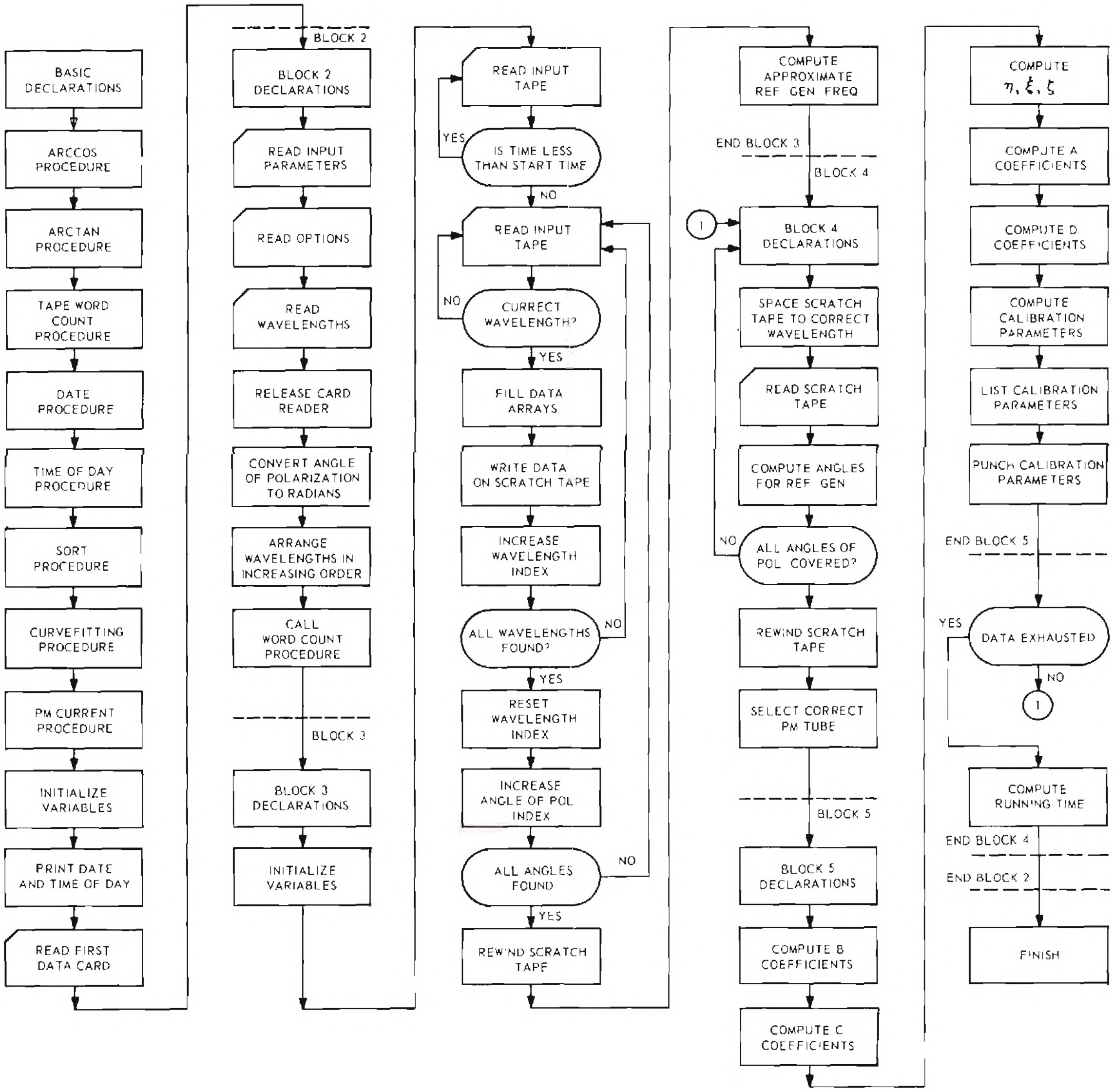

Figure 4. Calibration Program Flow Chart.

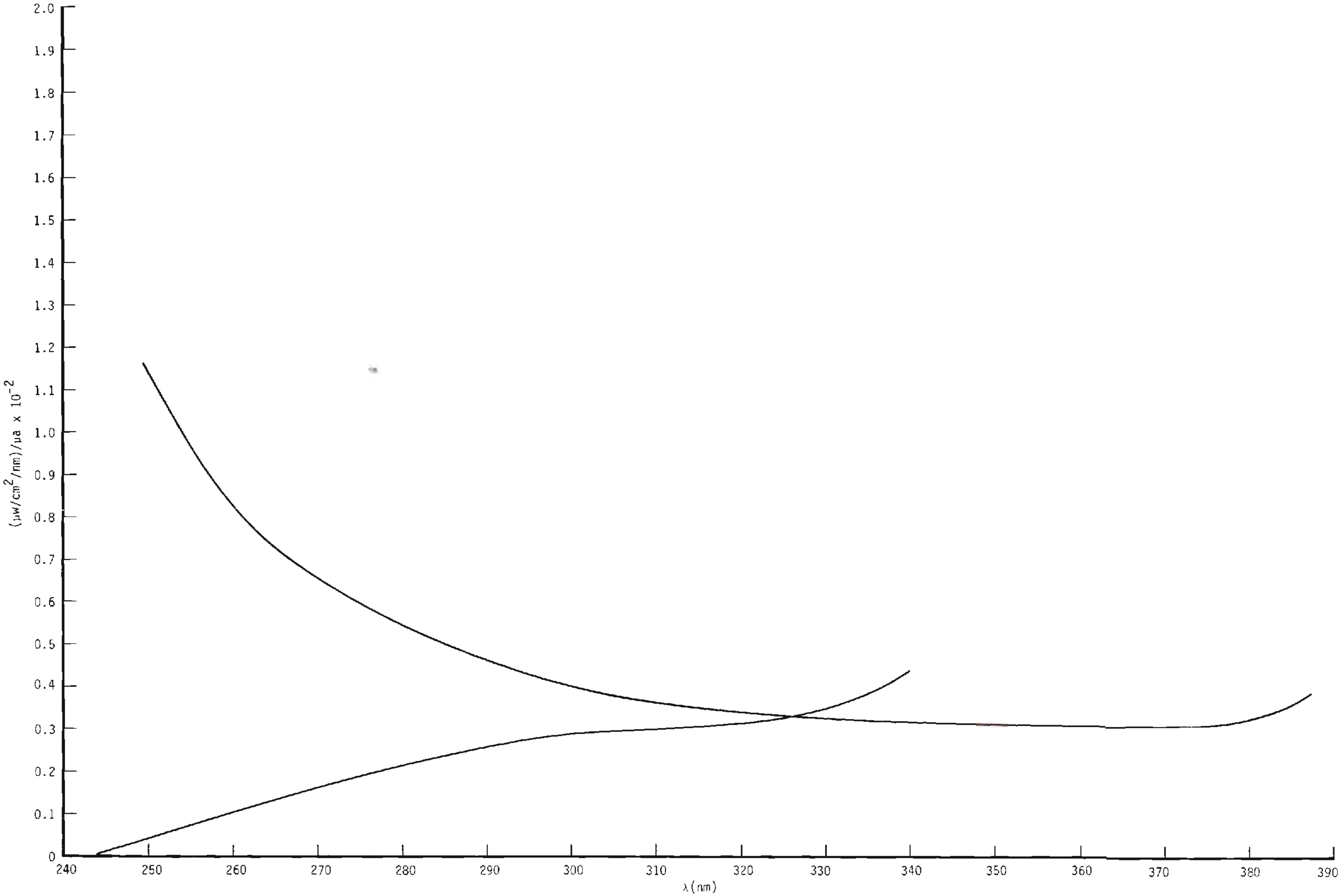

Figure 5. Plot of C Versus Wave Length.

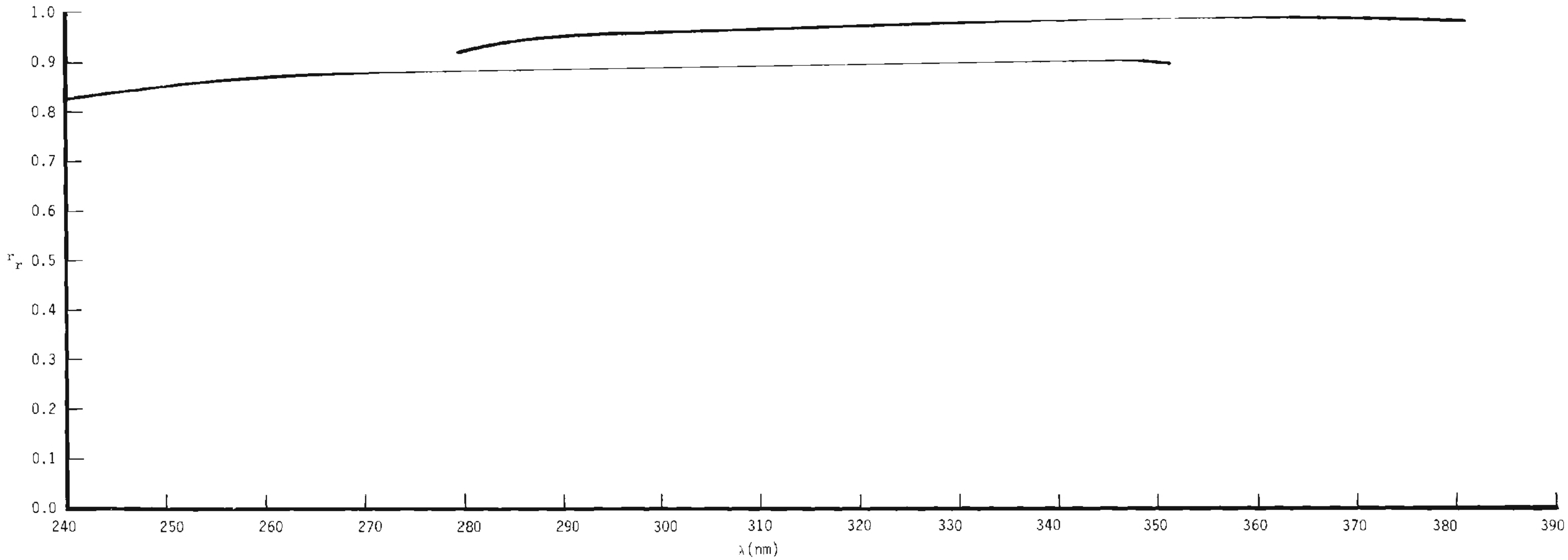

Figure 6. Plot of r_r Versus Wave Length.

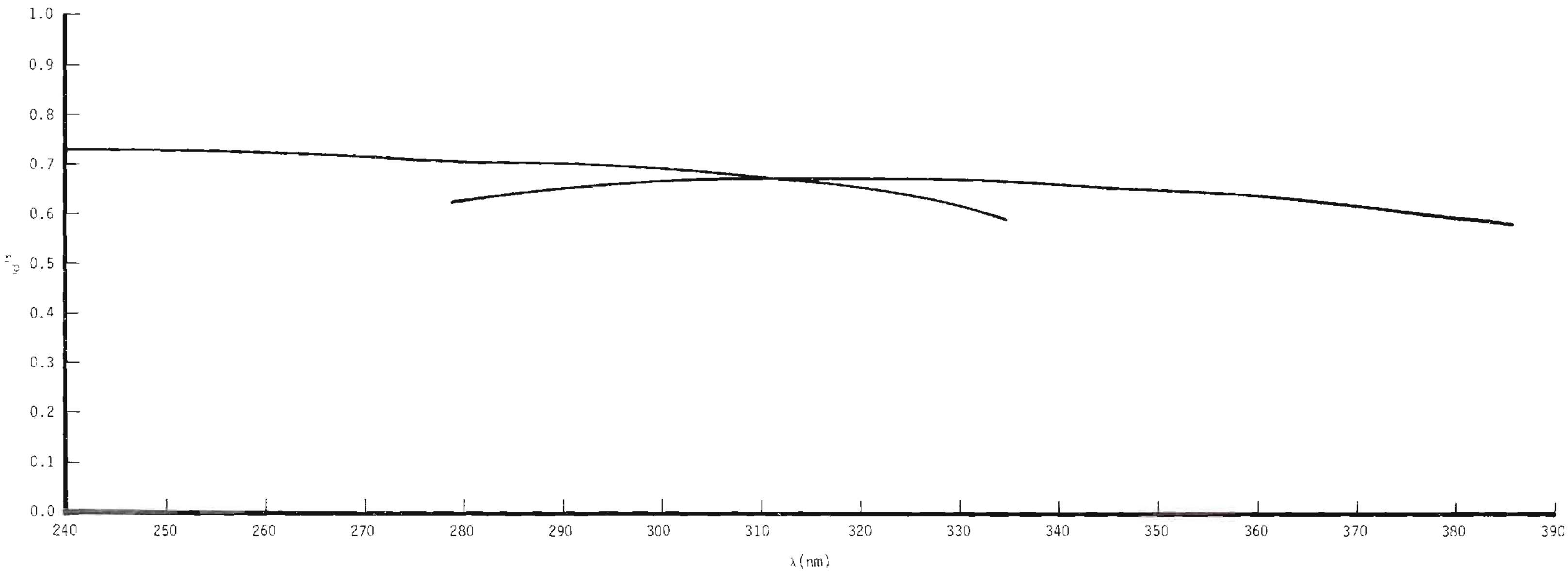

Figure 7. Plot of r_p Versus Wave Length.

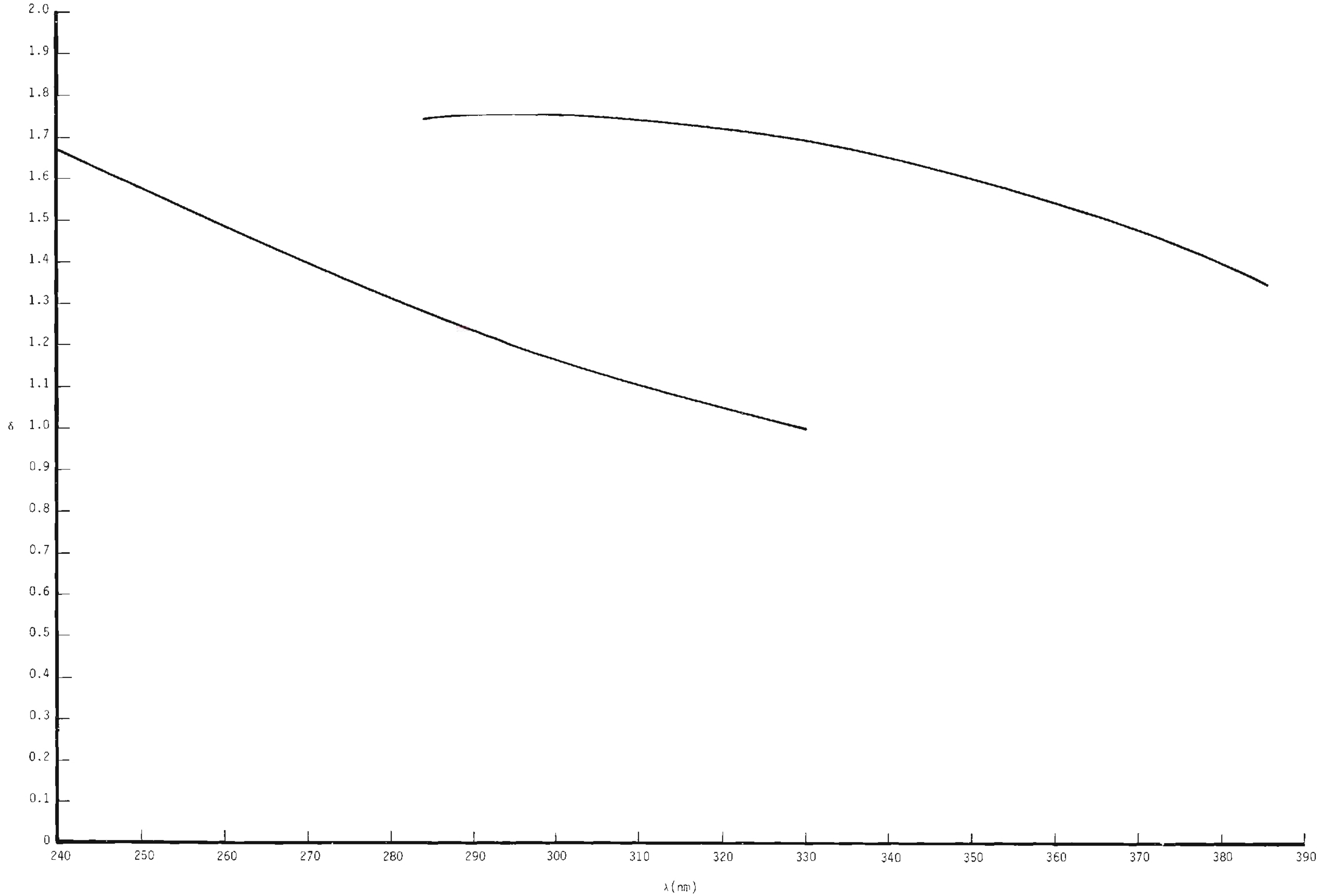

Figure 8. Plot of δ Versus Wave Length.

the coefficients $1/C(1 + r_p^2)(1 + r_r^2)$ in equation (4)] are calculated for various wave lengths and stored on an auxiliary input tape for the main reduction program. Thus the calibration parameters can be changed without having to change the main reduction program. Other data which are included on this input tape are scatter angle and altitude-time profile. After this auxiliary input tape is generated, the main portion of the data reduction can commence.

CHAPTER IV

DATA REDUCTION

Upon termination of the balloon flight, the data which have been recorded on analog tape recorders are processed through analog-to-digital converters (ADC). Since it is necessary to approximate the 40 cps AC generator signal during the computational process, the continuous data (FM data) is sampled at a rate of 500 samples per second per channel. The ADC processes are a service supplied with the balloon launch facilities, and therefore the digital tapes that are received are in a usable but not ideal condition to be used with computing equipment at Georgia Tech. The sampling limitation on the ADC is a total of 1000 samples per second. Therefore two tapes are produced for the four input channels—two photomultiplier tube signals, a grating position indicator, and the AC generator. The commutator data are digitized at a rate of one digital output value for each segment during a frame; with the commutator that was used, this rate was 70 samples per second. A commutator is a switching device that selects sequentially one of a number of inputs and connects it to the output; the commutator that was used had 28 inputs, and it sampled each of these input $2\frac{1}{2}$ times per second. A frame is one string of the 28 samples plus synchronization, and a segment is one input signal during a frame. These tapes produced by ADC are written in a packed 36 bit word which is incompatible with the 48 bit word used by the Georgia Tech Burroughs computer. Therefore a word length conversion is necessary.

A block diagram of the flow of information from the balloon package to the final results is shown in Figure 9. The calibration tapes are made a few days prior to the flight, using telemetry and under all practical conditions of a flight. Thus, the calibration includes any nonlinearity, frequency response and phase shift that might be present in the telemetry system. The altitude and temperature profiles are provided by the launch organization after the completion of the flight, and the altitude profile must be manually fed into the computer program so that the altitude at which the data are obtained can be printed with the data. As a consequence of the wide use of electronic data processing equipment, the amount of data which is handled manually is nominal.

The computer programs which perform these various stages are written in Algol-60 with the programming concept that they should be more general than specific in their application, i.e. if a foreseable change is made, this should not necessitate any major change in the program.

General Problems in Data Reduction

Every digital tape that is received from the ADC process must first undergo a word length conversion to make the tapes compatible with the Georgia Tech computer. Besides changing the word length from 36 to 48 bits, the converter program also provides an output of the converted data on a labeled tape to facilitate future handling. Two converter programs are used: one, the commutator converter program, converts the single commutator tape and produces an output tape in a similar format as the input tape, and the other, the FM data conversion program, converts and merges the two FM data input tapes into a single output tape. A third program is necessary to add wave length to the FM data output tape. The format

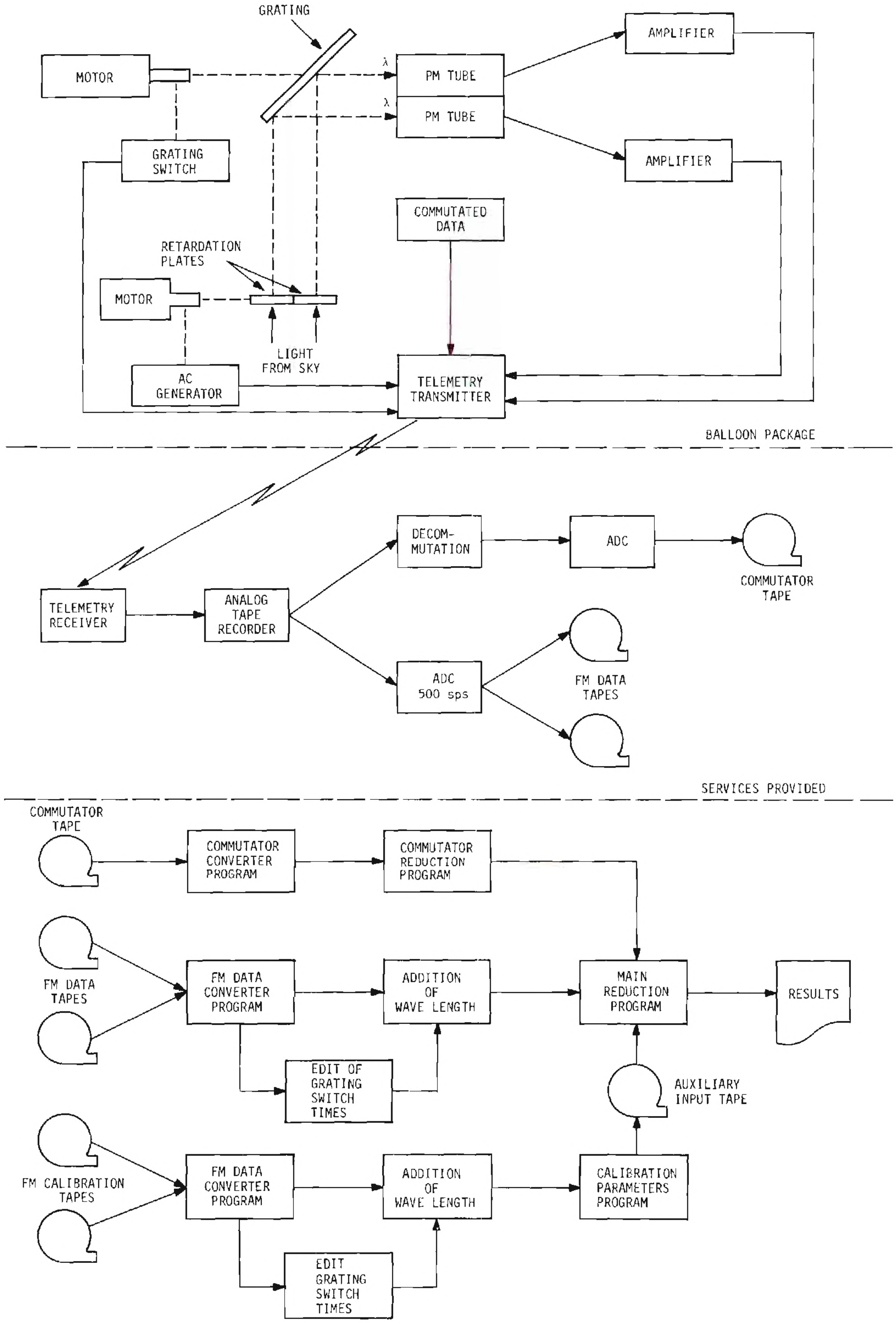

Figure 9. General Flow of Information.

for these tapes is given in Appendix C as well as a detailed explanation of the programs.

These three programs are as follows:

FM Data Converter Program

Basically this program converts the 36 bit digital tape into a digital tape which is usable on the Georgia Tech's Burroughs B-5500 computer. The tapes are in the same format except that the word length is 36 bits, while that of the B-5500 is 48 bits. The program accepts as input the two tapes containing the four digitized channels of FM/FM data and produces one tape containing all four channels. (At the present time, the digital input tapes are written in low density—200 bits per inch—on the tape, but this is not a limitation for the B-5500, which can read in either low or high density.) The flow diagram for this program is shown in Figure 10, while the reproduced Algol deck is shown in Appendix C.

Commutator Converter Program

This program differs slightly from the data conversion program in that it will only accept a single tape at a time as input. However, the basic logic is the same, and the output formats have similarities. The program converts the data tape generated by the ADC process, from a 36 bit word length to a 48 bit word length for use on the B-5500. The flow diagram is shown in Figure 11, while the reproduced Algol deck is presented in Appendix C. The program converts the word length and lists on the line printer periodically the tape block which is currently being written, and when the input data tape is exhausted, the output tape is rewound and periodically dumped to serve as a check on the conversion process by comparing this listing with the previous listing.

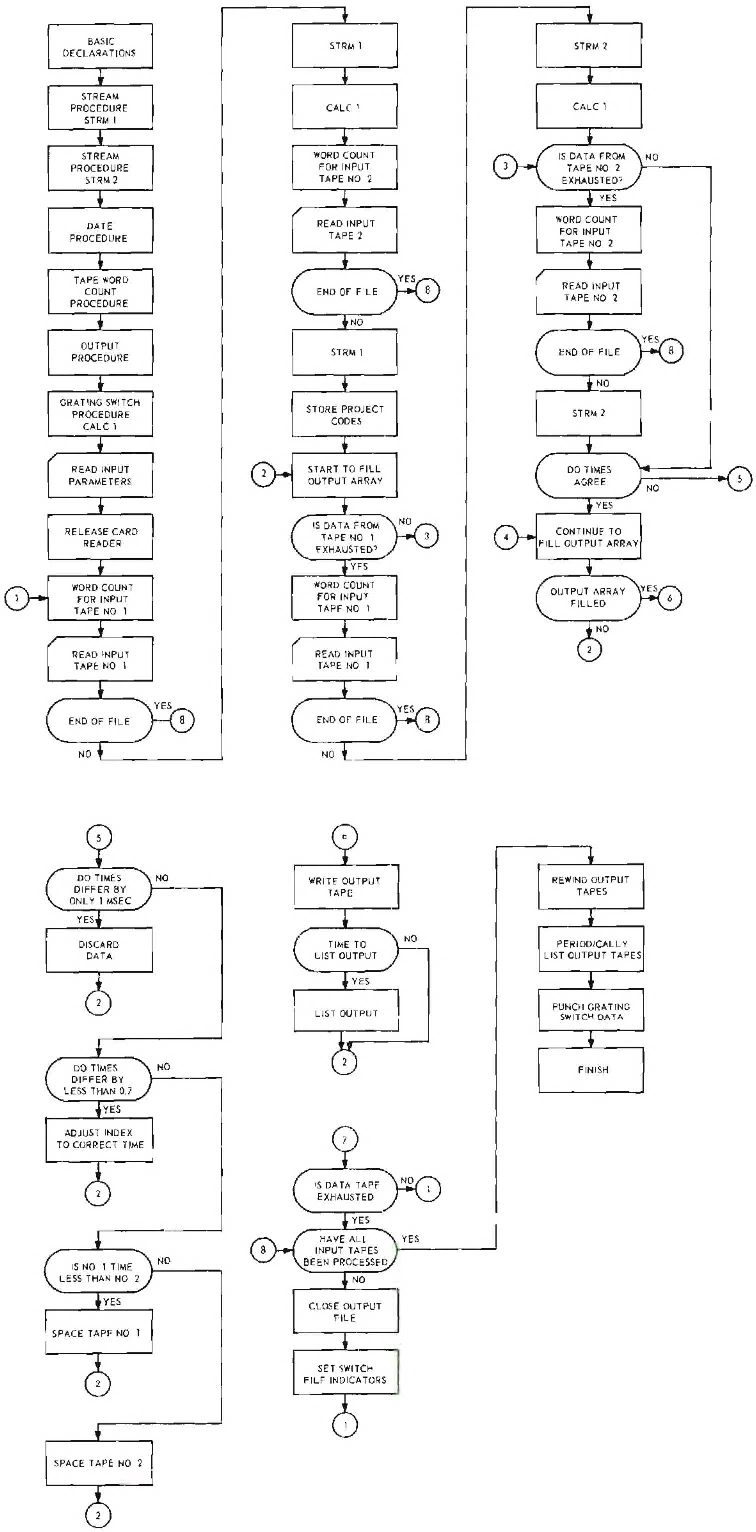

Figure 10. FM Data Converter Flow Chart.

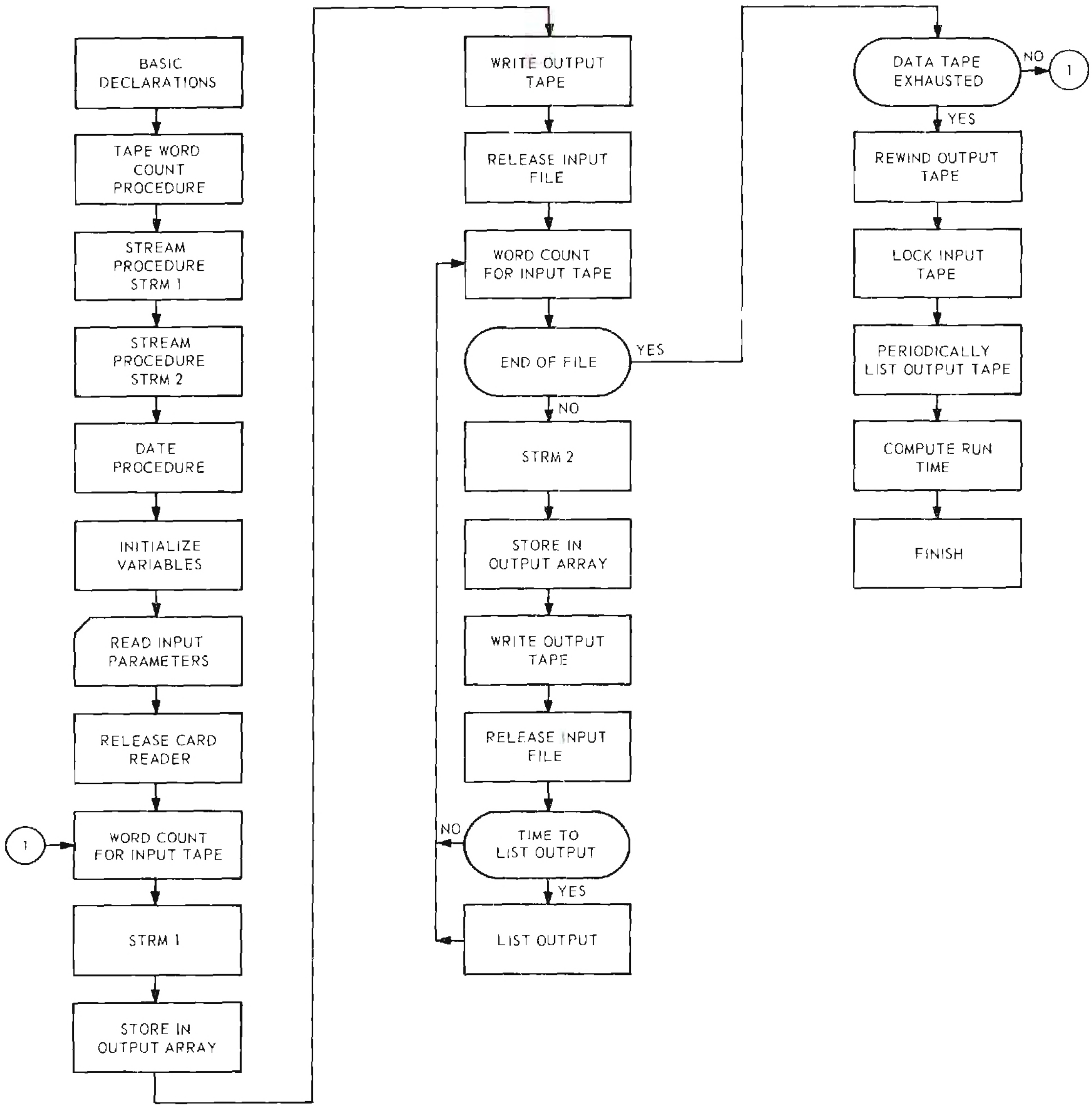

Figure 11. Commutator Converter Flow Chart.

Addition of Wave Length Program

The third program generates the input tape for the main data reduction program by providing the wave length at which a given data point was taken. During the FM data conversion process, the times at which the grating switch opened are punched onto cards. These times are then hand edited (to remove bad times and to include times which may have been missed by a malfunction of the grating switch) and used as input for the addition of wave length program. The output of the program is a tape with a similar format to the input tape (the output tape from the FM data conversion program), but with wave length added in bits 1 through 13. The flow diagram for this program is shown in Figure 12 while the reproduced Algol deck is in Appendix C.

Specific Problems in Data Reduction

After the completion of the addition of wave length to the FM data and conversion of the commutator data, two additional programs are needed to obtain the final answers. One, the commutator reduction program, reduces and lists the data that are contained on the commutator, such as scatter angle, battery voltages, error signals and temperatures. The other program obtains intensity and polarization parameters versus wave length. This second program also combines some of the results of the commutator reduction program so that the scatter angle and an "on-sun" flag are contained in the output. Each of these two programs will be discussed.

Commutator Reduction Program

The commutator reduction program uses the data from the commutator conversion program, averages it over a suitable time interval, converts the digital value to engineering units (degrees, volts, temperature, etc.),

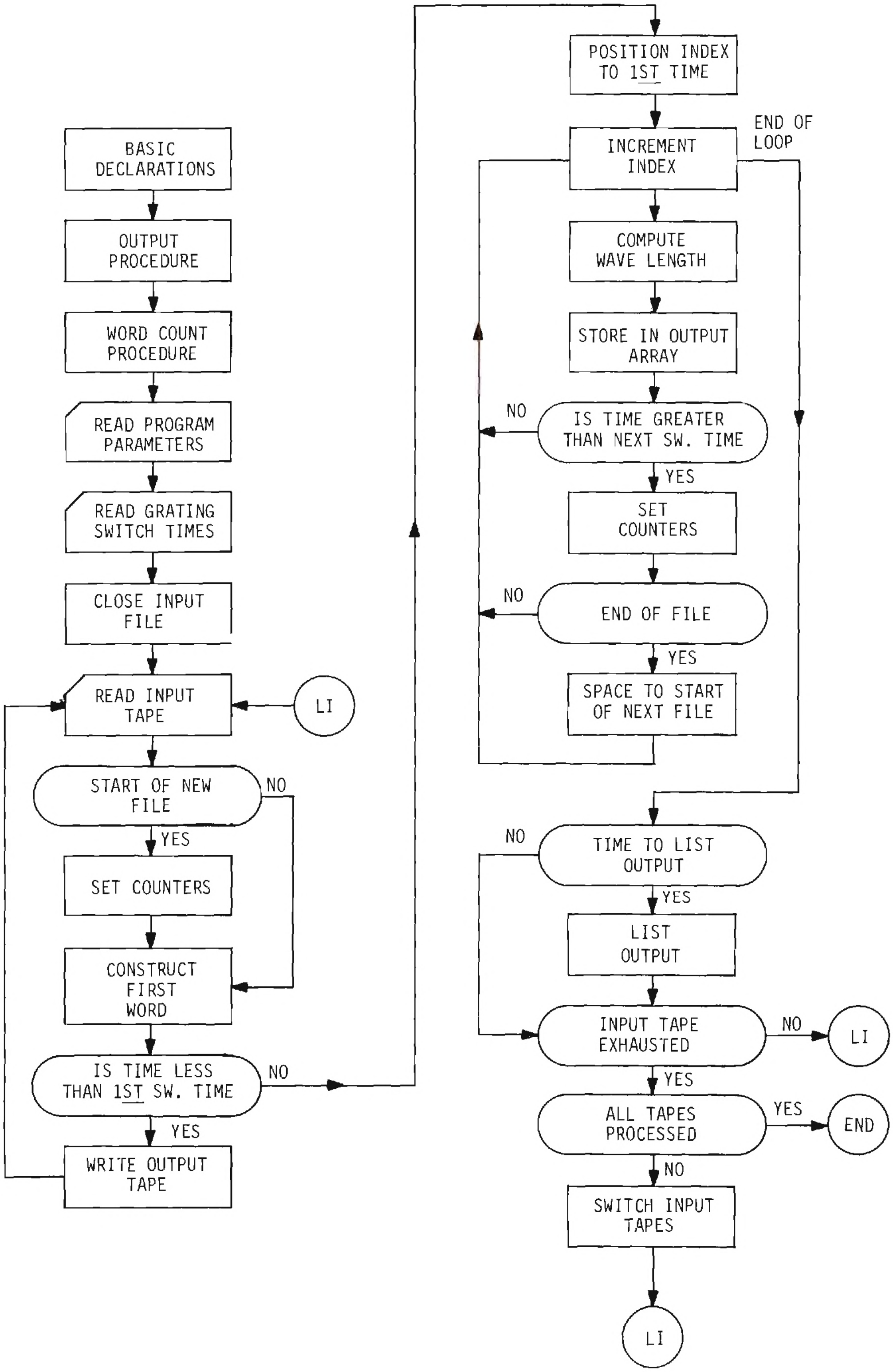

Figure 12. Addition of Wave Length Flow Chart.

and lists the output. For the 1 July 1965 flight, a list of commutator segments versus channel number (multiplex number, also) is given in Table 1. The maximum and the minimum signals (5.0 and 1.0 volts, respectively) are used for calibration purposes, and at present is the only calibration on the commutated data. The flow chart for this program is shown in Figure 13, while the Algol deck is reproduced in Appendix C.

Main Data Reduction Program

This final program in the chain produces the intensity and the modified Stokes parameters versus wave length for a specified time interval. A time correlation is used to obtain the altitude and scatter angle which have been derived elsewhere; the altitude is obtained from the launch organization, and the scatter angle is obtained from the commutator data. The program requires two input tapes: (1) the output tape from the addition of wave length program, and (2) an auxiliary input tape which contains certain constants which are used in the program, and also the temperature, altitude, and scatter angle profiles. Basically the program is similar to the calibration program, although it is not nearly as complex. The flow chart for this program is shown in Figure 14, while the reproduced Algol deck is contained in Appendix C.

This program has been used with a portion of the data obtained on the 1 July 1965 flight, and the results are shown graphically in Figures 15 and 16. Figure 15 shows the intensity of the scattered sunlight radiation at float altitude (121,000 feet) at approximately 71 degrees from the sun. Figure 16 shows the degree of polarization of the scattered sunlight under the same conditions. The calculation of the degree of polarization and the other modified Stokes parameters becomes inaccurate when the total

Table 1. Commutator Segments

Segment Number	Description
1	Maximum signal
2	Minimum signal
3	Solar monitor
4	Azimuth position
5	Elevation scan potentiometer
6	Pointing control Az CW error signal
7	Pointing control El CW error signal
8	Pointing control Az CCW error signal
9	Pointing control El CCW error signal
10	Electronics temperature
11	Optics temperature
12	Pointing control temperature
13	Ground plane temperature
14	Main battery temperature
15	TM battery temperature
16	Electronics battery temperature
17	TM battery voltage
18	Main battery voltage
19	Electronics battery voltage
20	Solar position photocall #1
21	Solar position photocall #2
22	Solar position photocall #3
23	Calibration light, electronics heater multiplex
24	Battery heater multiplex
25	Regulated 18 v. for pointing control
26	Camera pulse
27	Regulated 12 v. for thermistors
28	Reference voltage

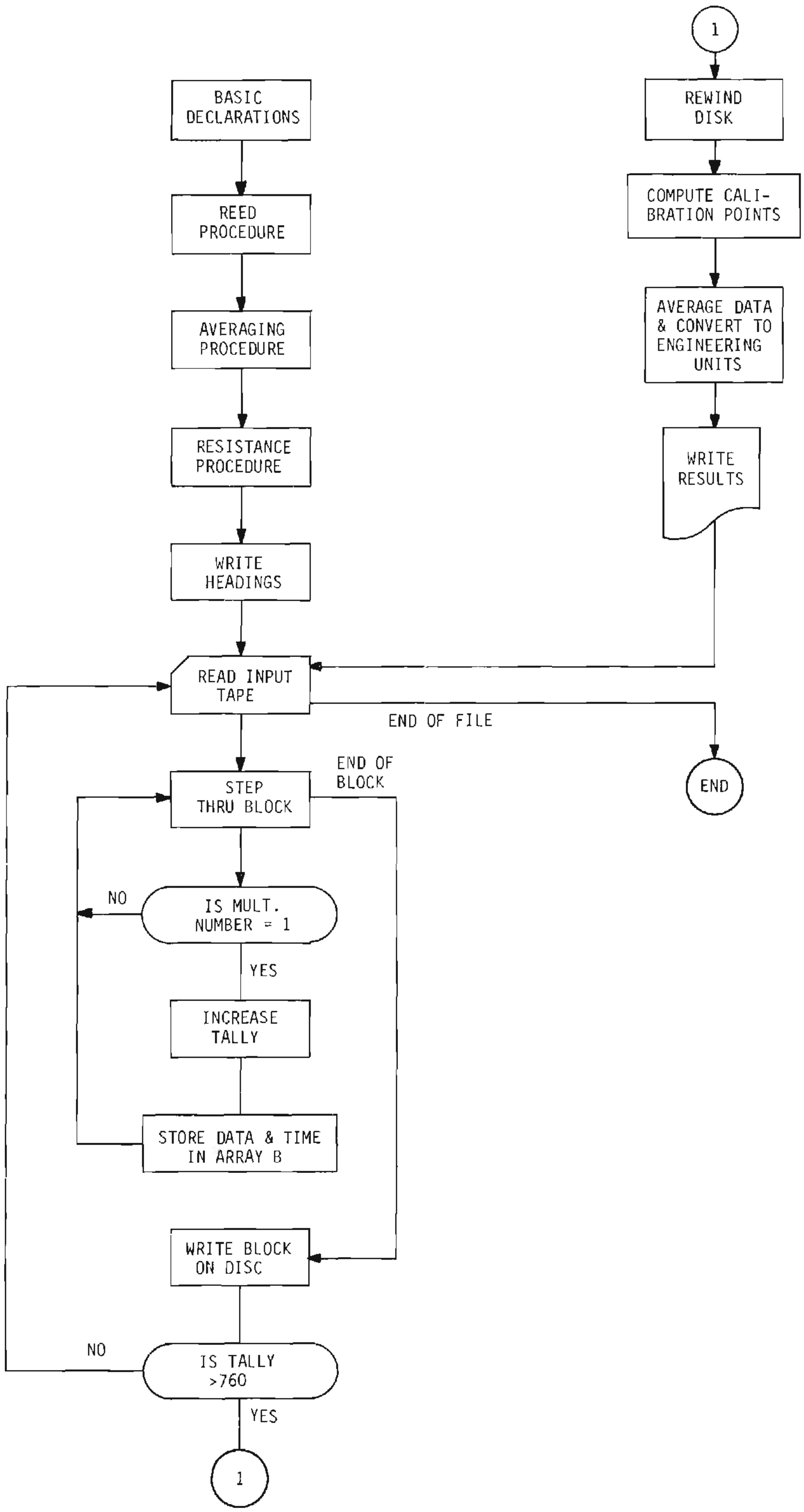

Figure 13. Commutator Reduction Flow Chart.

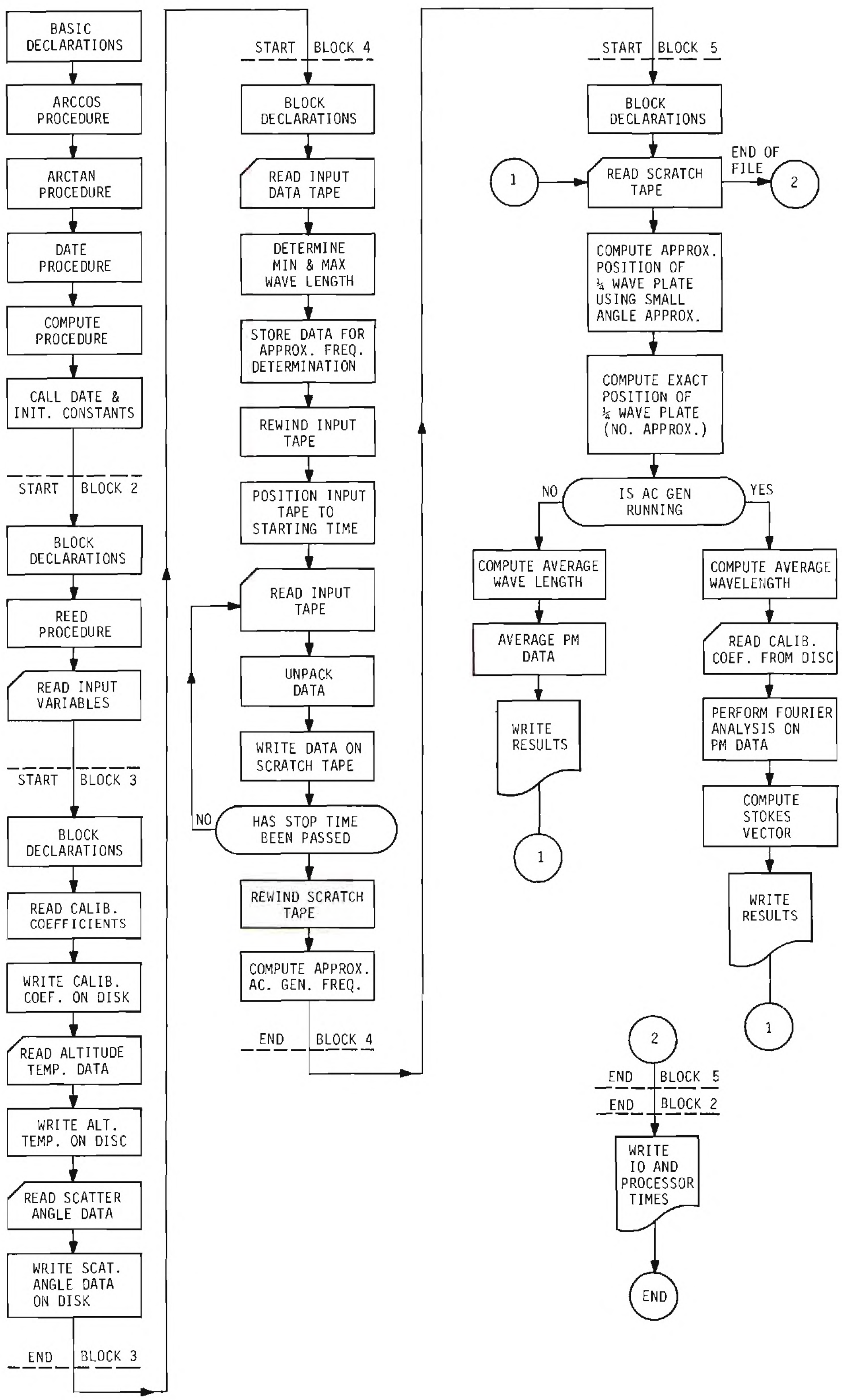

Figure 14. Main Data Reduction Flow Chart.

intensity is low, and hence, for those wave lengths where the intensity is low, the probable error on the other calculations increases. The ellipticity as determined by this method was never more than 0.03, and this figure could arise from computational errors and an inexact numerical approximation during the Fourier analysis of the photomultiplier tube signals. The angle of polarization was approximately horizontal. A detailed error evaluation has not been performed on the overall system, and therefore no estimate of probable error is available.

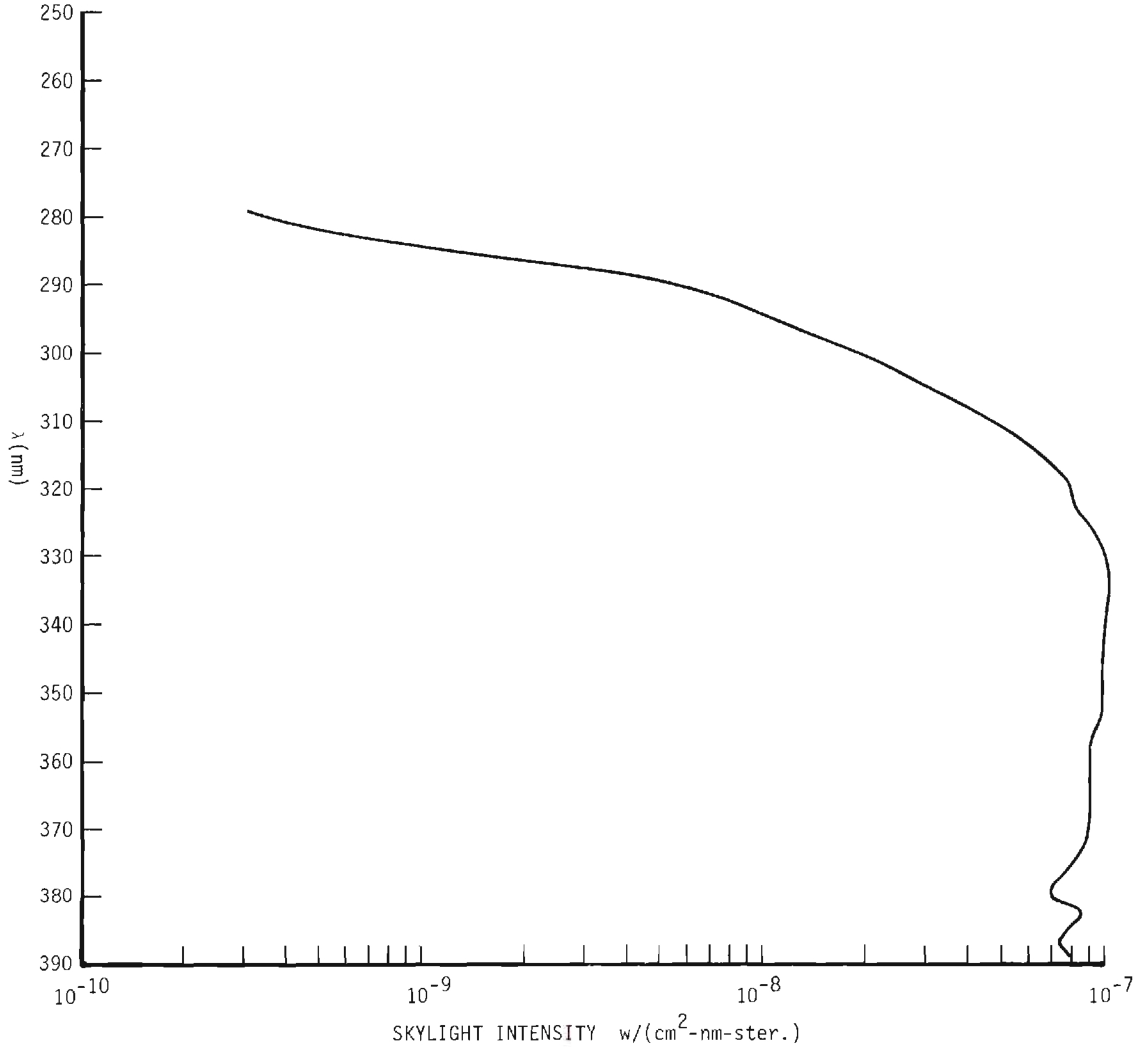

Figure 15. Spectral Intensity of a Portion of the UV Sky.

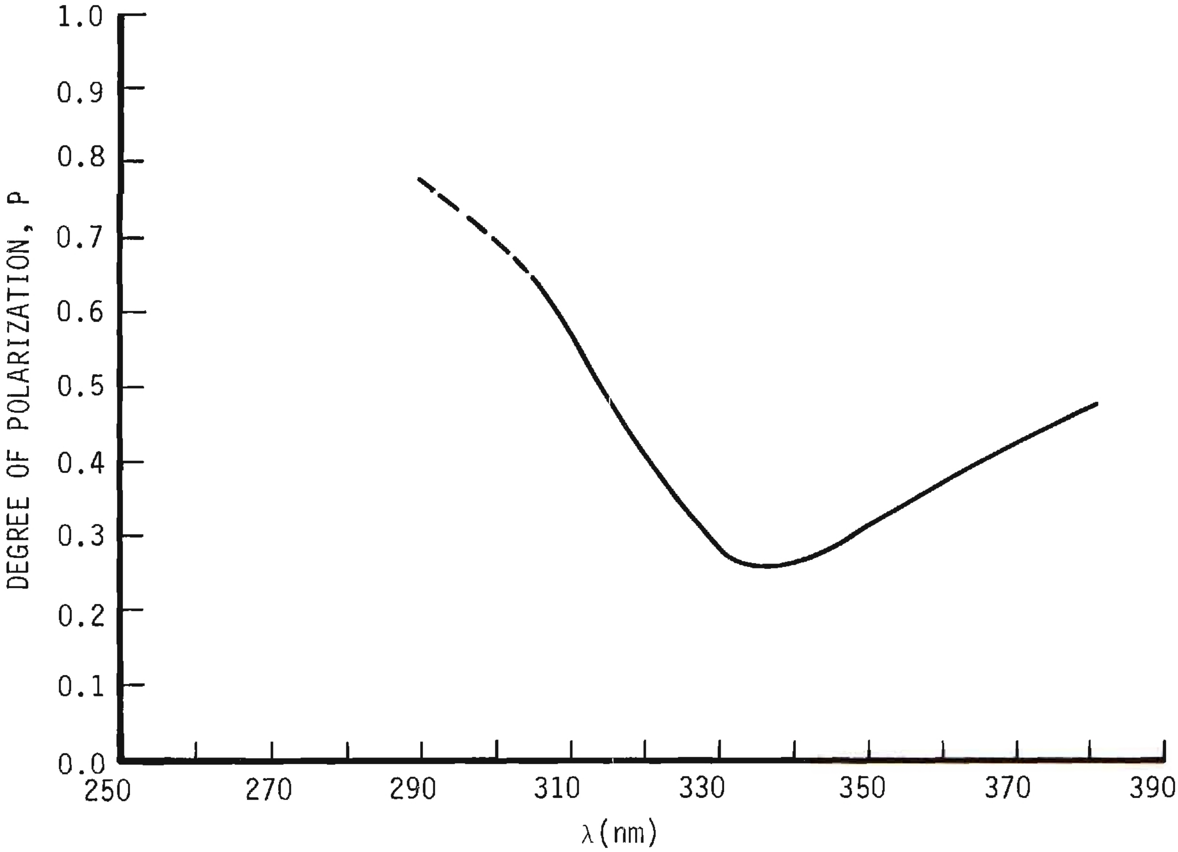

Figure 16. Degree of Polarization of a Portion of the UV Sky.

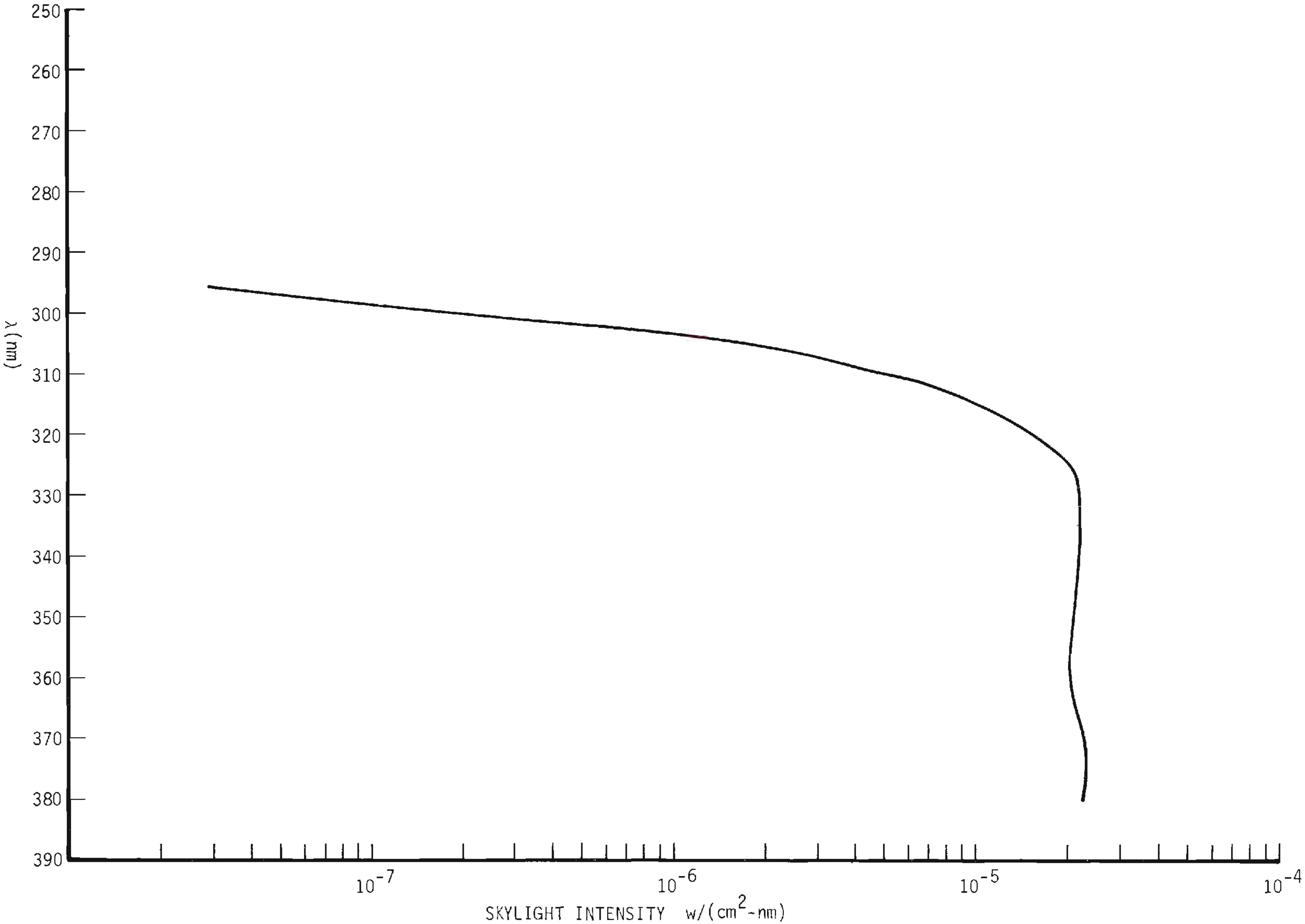

Figure 17. Spectral Intensity of the UV Sky Radiation.

CHAPTER V

SUMMARY AND CONCLUSIONS

The knowledge of the ultraviolet radiation incident from the sun and sky is of vital interest for geophysics as well as for biophysics. Considerable work has been done in the field from ground based observatories such as Davos, Switzerland, and the purpose of this experiment is to supplement these observations by taking data at high altitudes in a similar wave length region. The failure of the instrumentation package by the entanglement of the visibility flags on the 1 July 1965 flight prevented the acquisition of data from all scatter angles, but sufficient data were obtained to demonstrate the feasibility and practicality of the experiment. The results shown in Figures 15 and 16 agree with the experimental results of Bener[8] (Figure 17) and also some of the theoretical results of Sekera[3] regarding skylight intensity. It should be noted that Bener's data are taken for global sky radiation, whereas the data shown in Figures 15 and 16 are for an occluded angle of approximately 0.2 steradians. This agreement supports all phases of the experiment including calibration and data reduction, and it also shows that imperfect optical polarizing components can be successfully used in an Ebert-Fastie-Sekera arrangement to determine polarization and intensity.

Various improvements are possible to increase the accuracy, reliability, and performance of the overall system, although the basic design should remain intact. The data reduction programs can be improved and streamlined to reduce the amount of computer time necessary to reduce the data completely. An additional program could be written to perform the editing of the grating switch times prior to the addition of wave length. A program could also be written to provide a time edit of the data to eliminate undesirable data points which have an error in time. A master data preparation program could be assembled from programs already in existence, or suggested, that uses as input the tapes as received from the analog-to-digital conversion and produces an output tape with time editing which is suitable as input to the final data reduction programs. Work is currently being performed along this line.

The results presented here were not the prime objective of this thesis, but rather were used to demonstrate that answers could be obtained through the data reduction chain that has been presented within the text of the thesis. A careful evaluation of these data and future data should be made to ascertain overall accuracy and probable error. Since the data presented represent only one small portion of the sky, no definite conclusions can be reached regarding the mapping of the sky intensity, but the conclusion can be made that the data reduction system produces the desired results.

APPENDIX A

STOKES VECTORS AND MUELLER CALCULUS

Completely or partially polarized light can be specified by a set of four quantities known as the Stokes parameters, all of which have the dimensions of intensity.[9] These four parameters are usually represented by a column vector as:

$$\begin{bmatrix} I \\ Q \\ U \\ V \end{bmatrix} \quad . \tag{A-1}$$

The electromagnetic definitions of these quantities are

$$I = \langle e_x^2 + e_y^2 \rangle \tag{A-2a}$$

$$Q = \langle e_x^2 - e_y^2 \rangle \tag{A-2b}$$

$$U = \langle 2e_x e_y \cos \gamma \rangle \tag{A-2c}$$

$$V = \langle 2e_x e_y \sin \gamma \rangle \quad , \tag{A-2d}$$

where e_x and e_y are the instantaneous x and y components of the electric field respectively, γ is the relative phase angle between e_x and e_y, and the brackets indicate a time average. The first parameter, I, is to be interpreted as the total intensity of the light, and the remaining Stokes parameters lead to the following inequality:

$$I^2 \geq Q^2 + U^2 + V^2 \quad . \tag{A-3}$$

The intensity is equal to the square root of the sum of the other three only if the beam is completely polarized, i.e., no unpolarized component present. It is convenient to define the degree of polarization, P, as:

$$P = \frac{(Q^2 + U^2 + V^2)^{\frac{1}{2}}}{I} \tag{A-4}$$

If Φ is the angle which the major axis of the elliptically polarized light (as a special case, linearly polarized light) makes with the reference axis, then

$$\begin{aligned} I &= I \\ Q &= PI \cos 2\Phi \\ U &= PI \sin 2\Phi \\ V &= V \end{aligned} \tag{A-5}$$

and the quantities, I, P, V, and Φ will be designated as the modified Stokes parameters. These do not all have the same dimensions, and therefore will not transform as the normal Stokes vector.

In Mueller calculus,[10] an optical device is represented by a four by four matrix which operates on the Stokes vector of the incident light to give the Stokes vector of the transmitted (or reflected) light. For a non-perfect linear polarizer, let a and b be the fractional amplitude of transmission of the major and minor axis respectively (ideally, $a = 1$ and $b = 0$). The Mueller matrix for such a polarizer can be derived as follows: let m_{ij} be an element of the Mueller matrix M_P, and let it operate on completely unpolarized light.

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} m_{11} \\ m_{21} \\ m_{31} \\ m_{41} \end{bmatrix} \quad \text{(A-6)}$$

Now we expect the Stokes vector of the transmitted light to be (cf A-2)

$$\frac{1}{2} \begin{bmatrix} a^2 + b^2 \\ a^2 - b^2 \\ 0 \\ 0 \end{bmatrix} , \quad \text{(A-7)}$$

and hence

$$m_{11} = \tfrac{1}{2}(a^2 + b^2) \quad \text{(A-8a)}$$

$$m_{21} = \tfrac{1}{2}(a^2 - n^2) \quad \text{(A-8b)}$$

$$m_{31} = m_{41} = 0 \quad . \quad \text{(A-8c)}$$

(The factor of $\frac{1}{2}$ is for normalization). After making these substitutions into the Mueller matrix M_P, the matrix operates on light completely linearly polarized at 0° as follows:

$$\begin{bmatrix} \frac{1}{2}(a^2 + b^2) & m_{12} & m_{13} & m_{14} \\ \frac{1}{2}(a^2 - b^2) & m_{22} & m_{23} & m_{24} \\ 0 & m_{32} & m_{33} & m_{34} \\ 0 & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(a^2 + b^2) + m_{12} \\ \frac{1}{2}(a^2 - b^2) + m_{22} \\ m_{32} \\ m_{42} \end{bmatrix} . \quad \text{(A-9)}$$

Here, we expect the Stokes vector to be

$$\begin{bmatrix} a^2 \\ a^2 \\ 0 \\ 0 \end{bmatrix} , \tag{A-10}$$

so that

$$m_{12} = \tfrac{1}{2}(a^2 - b^2) \tag{A-11a}$$

$$m_{22} = \tfrac{1}{2}(a^2 + b^2) \tag{A-11b}$$

$$m_{32} = m_{42} = 0 \quad . \tag{A-11c}$$

After these substitutions have been made into the Mueller matrix M_P, the matrix operates on completely linearly polarized light at 45° as follows:

$$\tfrac{1}{2}\begin{bmatrix} a^2 + b^2 & a^2 - b^2 & 2m_{13} & 2m_{14} \\ a^2 - b^2 & a^2 + b^2 & 2m_{23} & 2m_{24} \\ 0 & 0 & 2m_{33} & 2m_{34} \\ 0 & 0 & 2m_{43} & 2m_{44} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \tfrac{1}{2}\begin{bmatrix} a^2 + b^2 + 2m_{13} \\ a^2 - b^2 + 2m_{23} \\ 2m_{33} \\ 2m_{43} \end{bmatrix} \tag{A-12}$$

Here we expect the transmitted vector to be

$$\begin{bmatrix} \tfrac{1}{2}(a^2 + b^2) \\ \tfrac{1}{2}(a^2 - b^2) \\ 2ab \\ 0 \end{bmatrix} . \tag{A-13}$$

Hence

$$m_{13} = m_{23} = m_{43} = 0 \tag{A-14a}$$

$$m_{33} = ab \quad . \tag{A-14b}$$

As before, after making these substitutions and letting the resultant matrix operate on right circularly polarized light, we obtain:

$$\frac{1}{2}\begin{bmatrix} a^2+b^2 & a^2-b^2 & 0 & 2m_{14} \\ a^2-b^2 & a^2+b^2 & 0 & 2m_{24} \\ 0 & 0 & 2ab & 2m_{34} \\ 0 & 0 & 0 & 2m_{44} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} a^2+b^2+2m_{14} \\ a^2-b^2+2m_{24} \\ 2m_{34} \\ 2m_{44} \end{bmatrix} \tag{A-15}$$

Here we expect the Stokes vector of the transmitted light to be

$$\begin{bmatrix} \frac{1}{2}(a^2+b^2) \\ \frac{1}{2}(a^2-b^2) \\ 0 \\ 2ab \end{bmatrix} \tag{A-16}$$

and hence

$$m_{14} = m_{24} = m_{34} = 0 \tag{A-16a}$$

$$m_{44} = ab \quad . \tag{A-16b}$$

After making these substitutions, we divide the entire matrix M_p by a^2, let $C_p = 1/2a^2$ and $r_p = b/a$ and obtain

$$M_p = C_p \begin{bmatrix} 1+r_p^2 & 1-r_p^2 & 0 & 0 \\ 1-r_p^2 & 1+r_p^2 & 0 & 0 \\ 0 & 0 & 2r_p & 0 \\ 0 & 0 & 0 & 2r_p \end{bmatrix} , \tag{A-17}$$

where r_p is then the ratio of the amplitude of transmission of the minor to the major axes, and C_p is a normalization constant.

For an imperfect retardation plate of retardance δ which shows some linear polarization and is oriented with its fast axis along a reference axis (0°), we can derive the Mueller matrix as follows.

For the case of unpolarized incident light and light polarized at 0°, since there is no y component of the electric field to produce a relative phase shift, the development of the Mueller matrix M_R follows that for a linear polarizer thus:

$$M_R = \frac{1}{2}\begin{bmatrix} a^2 + b^2 & a^2 - b^2 & 2m_{13} & 2m_{14} \\ a^2 - b^2 & a^2 + b^2 & 2m_{23} & 2m_{24} \\ 0 & 0 & 2m_{33} & 2m_{34} \\ 0 & 0 & 2m_{43} & 2m_{44} \end{bmatrix} . \qquad \text{(A-18)}$$

Letting this matrix operate on light polarized at 45° yields

$$\frac{1}{2}\begin{bmatrix} a^2 + b^2 & a^2 - b^2 & 2m_{13} & 2m_{14} \\ a^2 - b^2 & a^2 + b^2 & 2m_{23} & 2m_{24} \\ 0 & 0 & 2m_{33} & 2m_{34} \\ 0 & 0 & 2m_{43} & 2m_{44} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} a^2 + b^2 + 2m_{13} \\ a^2 - b^2 + 2m_{23} \\ 2m_{33} \\ 2m_{43} \end{bmatrix} . \qquad \text{(A-19)}$$

From the electromagnetic definition of the Stokes vector, we would expect the transmitted light to appear as

$$\begin{bmatrix} \frac{1}{2}(a^2 + b^2) \\ \frac{1}{2}(a^2 - b^2) \\ 2ab \cos \delta \\ 2ab \sin \delta \end{bmatrix} , \qquad \text{(A-20)}$$

and hence

$$m_{13} = m_{23} = 0 \qquad \text{(A-21a)}$$

$$m_{33} = ab \cos \delta \qquad \text{(A-21b)}$$

$$m_{43} = ab \sin \delta \quad . \qquad \text{(A-21c)}$$

After these substitutions are made in the Mueller matrix M_R, the matrix operates on right circularly polarized light to yield

$$\frac{1}{2}\begin{bmatrix} a^2+b^2 & a^2-b^2 & 0 & 2m_{14} \\ a^2-b^2 & a^2+b^2 & 0 & 2m_{24} \\ 0 & 0 & 2ab\cos\delta & 2m_{34} \\ 0 & 0 & 2ab\sin\delta & 2m_{44} \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} a^2+b^2+2m_{14} \\ a^2-b^2-2m_{24} \\ 2m_{34} \\ 2m_{44} \end{bmatrix} \qquad \text{(A-22)}$$

Again from the electromagnetic definition, we would expect the Stokes vector of the transmitted light to be

$$\begin{bmatrix} \frac{1}{2}(a^2+b^2) \\ \frac{1}{2}(a^2-b^2) \\ 2ab\cos(90+\delta) \\ 2ab\sin(90+\delta) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(a^2+b^2) \\ \frac{1}{2}(a^2-b^2) \\ -2ab\sin\delta \\ 2ab\cos\delta \end{bmatrix} . \qquad \text{(A-23)}$$

Hence

$$m_{14} = m_{24} = 0 \qquad \text{(A-24a)}$$

$$m_{34} = -ab\sin\delta \qquad \text{(A-24b)}$$

$$m_{44} = ab\cos\delta . \qquad \text{(A-24c)}$$

After making these final substitutions, we divide the entire matrix by a^2, let $C_r = 1/2a^2$ and $r_r = b/a$ and obtain

$$M_R = C_r\begin{bmatrix} 1+r_r^2 & 1-r_r^2 & 0 & 0 \\ 1-r_r^2 & 1+r_r^2 & 0 & 0 \\ 0 & 0 & 2r_r\cos\delta & -2r_r\sin\delta \\ 0 & 0 & 2r_r\sin\delta & 2r_r\cos\delta \end{bmatrix} . \qquad \text{(A-25)}$$

This is the Mueller matrix for an imperfect retardation plate with its fast axis along the reference axis. It is then a simple matter to obtain the matrix for an orbitrary angle β by operating on (A-25) as follows:

$$M_{R_\beta} = R\,[2\beta]\,[M_R]\,R\,[-2\beta] \quad , \qquad \text{(A-26)}$$

where R is the well known rotation matrix

$$R[\alpha] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(A-27)}$$

Upon completion of this operation, the following is obtained:

$$
[M_{R_\beta}] = C_r \begin{bmatrix}
1 + r_r^{\,2} & \left(1 - r_r^{\,2}\right) \cos 2\beta & \left(1 - r_r^{\,2}\right) \sin 2\beta & 0 \\
\left(1 - r_r^{\,2}\right) \cos 2\beta & \begin{array}{l} \frac{1}{2}\left(1 + r_r^{\,2} + 2r_r \cos \delta\right) + \\ \frac{1}{2}\left(1 + r_r^{\,2} - 2r_r \cos \delta\right) \cos 4\beta \end{array} & \frac{1}{2}\left(1 + r_r^{\,2} - 2r_r \cos \delta\right) \sin 4\beta & 2r_r \sin \delta \sin 2\beta \\
\left(1 - r_r^{\,2}\right) \sin 2\beta & \frac{1}{2}\left(1 + r_r^{\,2} - 2r_r \cos \delta\right) \sin 4\beta & \begin{array}{l} \frac{1}{2}\left(1 + r_r^{\,2} + 2r_r \cos \delta\right) - \\ \frac{1}{2}\left(1 + r_r^{\,2} - 2r_r \cos \delta\right) \cos 4\beta \end{array} & -2r_r \sin \delta \cos 2\beta \\
0 & -\,2r_r \sin \delta \sin 2\beta & 2r_r \sin \delta \cos 2\beta & 2r_r \cos \delta
\end{bmatrix}, \qquad \text{(A-28)}
$$

where β is the angle the fast axis makes with the reference axis, δ is the retardance of the phase plate, r_r is the ratio of the amplitudes of transmission of the fast to the slow axes, and C_r is a normalization constant.

APPENDIX B

DERIVATION OF THE CALIBRATION FORMULAE

After the operation indicated in equation (2) is performed, the intensity, I_A, of the light leaving the linear polarizer is detected by a photomultiplier tube, and this intensity is simply the first component of [A]. The signal out of the photomultiplier tube can be separated into components of β, the angle of rotation of the phase plate, as follows:

$$I_A = a_1 + a_2 \sin 2\beta + a_3 \cos 2\beta + a_4 \sin 4\beta + a_5 \cos 4\beta \ , \qquad \text{(B-1)}$$

where

$$a_1 = CI \left[(1 + r_p^2)(1 + r_r^2) + \tfrac{1}{2}(1 - r_p^2)(1 + r_r^2 + 2r_r \cos \delta) \ P \cos 2\phi\right] \qquad \text{(B-2a)}$$

$$a_2 = CI \left[(1 + r_p^2)(1 - r_r^2) \ P \sin 2\phi + 2Vr_r \ (1 - r_p^2) \ \sin \delta/I\right] \qquad \text{(B-2b)}$$

$$a_3 = CI \left[(1 - r_p^2)(1 - r_r^2) + (1 + r_p^2)(1 - r_r^2) \ P \cos 2\phi\right] \qquad \text{(B-2c)}$$

$$a_4 = \tfrac{1}{2}CI \left[(1 - r_p^2)(1 + r_r^2 - 2r_r \cos \delta) \ P \sin 2\phi\right] \qquad \text{(B-2d)}$$

$$a_5 = \tfrac{1}{2}CI \left[(1 - r_p^2)(1 + r_r^2 - 2r_r \cos \delta) \ P \cos 2\phi\right] \qquad \text{(B-2e)}$$

If the angle β is not known, but a relative angle, θ, is known, then

$$\beta = \theta + \zeta \qquad \text{(B-3)}$$

and I_A can be expanded in terms of θ in the form

$$I_A = b_1 + b_2 \sin 2\theta + b_3 \cos 2\theta + b_4 \sin 4\theta + b_5 \cos 4\theta \ , \qquad \text{(B-4)}$$

where

$$b_1 = a_1 \qquad \text{(B-5a)}$$

$$b_2 = a_2 \cos 2\zeta - a_3 \sin 2\zeta \qquad \text{(B-5b)}$$

$$b_3 = a_2 \sin 2\zeta + a_3 \cos 2\zeta \qquad \text{(B-5c)}$$

$$b_4 = a_4 \cos 4\zeta - a_5 \sin 4\zeta \qquad \text{(B-5d)}$$

$$b_5 = a_4 \sin 4\zeta + a_5 \cos 4\zeta \quad . \qquad \text{(B-5e)}$$

A Fourier analysis in terms of ϕ is now attempted on the b's. As before, if the angle ϕ is not known, but a relative angle, ψ, is known, then

$$\phi = \psi + \eta \quad , \qquad \text{(B-6)}$$

and equation (B-5) can be placed in the following form:

$$b_1 = c_{11} + c_{12} \sin 2\psi + c_{13} \cos 2\psi \qquad \text{(B-7a)}$$

$$b_2 = c_{21} + c_{22} \sin 2\psi + c_{23} \cos 2\psi \qquad \text{(B-7b)}$$

$$b_3 = c_{31} + c_{32} \sin 2\psi + c_{33} \cos 2\psi \qquad \text{(B-7c)}$$

$$b_4 = c_{42} \sin 2\psi + c_{43} \cos 2\psi \qquad \text{(B-7d)}$$

$$b_5 = c_{52} \sin 2\psi + c_{53} \cos 2\psi \quad , \qquad \text{(B-7e)}$$

where

$$c_{11} = d_1 \qquad \text{(B-8a)}$$

$$c_{12} = -d_2 \sin 2\eta \qquad \text{(B-8b)}$$

$$c_{13} = d_2 \cos 2\eta \qquad \text{(B-8c)}$$

$$c_{21} = d_3 \cos 2\zeta - d_5 \sin 2\zeta \qquad \text{(B-8d)}$$

$$c_{22} = d_4 \cos 2\xi_1 = d_4 \cos 2(\eta - \zeta) \tag{B-8e}$$

$$c_{23} = d_4 \sin \ \xi_1 = d_4 \sin 2(\eta - \zeta) \tag{B-8f}$$

$$c_{31} = d_3 \sin 2\zeta + d_5 \cos 2\zeta \tag{B-8g}$$

$$c_{32} = d_4 \sin 2\xi_1 = d_4 \sin 2(\eta - \zeta) \tag{B-8h}$$

$$c_{33} = -d_4 \cos 2\xi_1 = -d_4 \sin 2(\eta - \zeta) \tag{B-8i}$$

$$c_{42} = d_6 \cos 2\xi_2 = d_6 \cos 2(\eta - 2\zeta) \tag{B-8j}$$

$$c_{43} = d_6 \sin 2\xi_2 = d_6 \sin 2(\eta - 2\zeta) \tag{B-8k}$$

$$c_{52} = d_6 \sin 2\xi_2 = d_6 \sin 2(\eta - 2\zeta) \tag{B-8m}$$

$$c_{53} = -d_6 \cos 2\xi_2 = d_6 \cos 2(\eta - 2\zeta) \quad , \tag{B-8n}$$

where

$$d_1 = CI(1 + r_p^{\,2})(1 + r_r^{\,2}) \tag{B-9a}$$

$$d_2 = \tfrac{1}{2}CI(1 - r_p^{\,2})(1 + r_r^{\,2} + 2r_r \cos \delta) \ P \tag{B-9b}$$

$$d_3 = 2VCIr_r \ (1 - r_p^{\,2}) \sin \delta \tag{B-9c}$$

$$d_4 = CI(1 + r_p^{\,2})(1 - r_r^{\,2}) \ P \tag{B-9d}$$

$$d_5 = CI(1 - r_r^{\,2})(1 - r_p^{\,2}) \tag{B-9e}$$

$$d_6 = \tfrac{1}{2}CI(1 - r_p^{\,2})(1 + r_r^{\,2} - 2r_r \cos \delta) \ P \quad . \tag{B-9f}$$

At a given wave length, I_A is determined for at least 25 different β. The plane of incident polarization, ψ, is then rotated by a fixed amount (in this case, 11.25°), and I_A is again measured for at least 25 different β, and this process is repeated until the plane of incident polarization has passed through at least 180°. Then, for each different angle of incident polarization, the b's [equation (B-4)] are determined by a curve fitting process which produces a series of b_1's, say, as a function of ψ as expressed in equation (B-7a). Another multiple linear regression is then performed on say, b_1 to determine c_{11}, c_{12}, c_{13}, and after all the b's have been expanded in components of ψ, then all the c's would have been determined. It then follows from equation (B-8) that:

$$\eta = \frac{1}{2}\tan^{-1}\left(\frac{-c_{12}}{c_{13}}\right) \qquad \text{(B-10)}$$

$$\xi_1 = \frac{1}{2}\tan^{-1}\left(\frac{c_{23}}{c_{22}}\right) = \frac{1}{2}\tan^{-1}\left(\frac{c_{32}}{-c_{33}}\right) \qquad \text{(B-11)}$$

$$\xi_2 = \frac{1}{2}\tan^{-1}\left(\frac{c_{43}}{c_{42}}\right) = \frac{1}{2}\tan^{-1}\left(\frac{c_{52}}{-c_{53}}\right) \qquad \text{(B-12)}$$

$$\zeta = \eta - \xi_1 = \tfrac{1}{2}(\eta - \xi_2) \qquad \text{(B-13)}$$

Although it is possible to obtain the calibration parameters from the explicit form for the c's, it is somewhat more straight forward if the defining relationships of (B-3) and (B-6) are used, and the process is repeated by expanding I_A first in components of β as in (B-1) and then the a coefficients in terms of ϕ as

$$a_1 = d_1 + d_2 \cos 2\phi \qquad \text{(B-14a)}$$

$$a_2 = d_3 + d_4 \sin 2\phi \qquad \text{(B-14b)}$$

$$a_3 = d_5 + d_4 \cos 2\phi \qquad \text{(B-14c)}$$

$$a_4 = d_6 \sin 2\phi \qquad \text{(B-14d)}$$

$$a_5 = d_6 \cos 2\phi \quad . \qquad \text{(B-14e)}$$

From these relationships, the calibration parameters can be determined as follows:

$$r_r = \sqrt{\frac{1-k}{1+k}} \qquad \text{(B-15)}$$

$$\cos\delta = \frac{d_2 - d_6}{d_2 + d_6}\left(\frac{1 + r_r^2}{2r_r}\right) \tag{B-16}$$

$$r_p = \sqrt{\frac{kd_1 - d_5}{kd_1 + d_5}} \tag{B-17}$$

$$CI = \frac{d_1(1 + k)}{2(1 + r_p^2)} \quad , \tag{B-18}$$

where

$$k = \frac{1 - r_r^2}{1 + r_r^2} = \sqrt{\frac{d_4 d_5}{d_1(d_2 + d_6)}} \tag{B-19}$$

This effectively completes the optical calibration of the spectrophotopolarimeter.

APPENDIX C

COMPUTER PROGRAMS

This section contains the various computer programs that have been used in order to obtain data from the flight. It is assumed that the analog-to-digital conversion has been performed, and that the information is on magnetic tape. At present, six programs are necessary to completely reduce the data, and they are the FM Data Conversion, Commutator Conversion, Addition of Wave Length, Calibration Procedure, Commutator Reduction, and Main FM Reduction. Each of these will be discussed in detail.

FM Data Conversion Program

This program produces a tape which is compatible with the Burroughs B-5500 computer containing the FM data from all four channels—AC generator, grating switch, and the two photomultiplier tubes. The program also determines the times at which the grating switch opened, and these times are punched and listed as additional output. The program also lists the output tape periodically.

The program has as its input:

(1) FILE FIL1, in free field format, whether this is a test or run, the interval between records which are listed, the number or records at each interval to be listed, the number of pairs of input tapes (maximum 3), and the number of blocks or records to be converted (ignored if it is a production run),

(2) FILE FFMD1, tape unit unlabeled tape, the digital tape containing the AC generator and grating switch data,

(3) FILE FFMD2, tape unit, unlabeled tape, the digital tape containing the two photomultiplier tube signals.

(4) FILE FFMD3, same as (2)

(5) FILE FFMD4, same as (3)

(6) FILE FFMD5, same as (2)

(7) FILE FFMD6, same as (3)

Numbers (4) through (7) are only present if the number of pairs of input tapes is greater than one.

The program has as its output:

(1) FILE FIL2, line printer, a listing of the data as it is converted, and then after the tape is rewound, a listing at the proper interval of the output tape,

(2) FILE ZZZZZ01, labeled tape, a digital tape containing data, time, and multiplex number in a packed-word concept from the 1st pair of input tapes,

(3) FILE ZZZZZ02, same as (2) except from the 2nd pair of input tapes,

(4) FILE ZZZZZ03, same as (2) except from the 3rd pair of input tapes.

One block on the output tape contains one-half second of data. This arrangement permits easy positioning of the tape at a desired time. The data is stored on the output tape and the input tape in what is known as a packed-word, i.e., one computer word contains more than one word of information. The tapes as they are received are written in the following format:

Bits	0-10	Datum
	11-18	Time in seconds (maximum of 255)
	19-28	Time in milliseconds
	29-35	Multiplex number.

The datum which appears in the packed-word is the digital value in arbitrary units of the given multiplexer at the given time. The multiplex number is a number assigned by the ADC process to designate from which subcarrier the data were obtained. (On the commutated data tape, this number designates the segment from which the data are obtained.)

The output tape(s) are written in the following format:

Bits	0	Unused
	1-13	Wave length (to be added later)
	14-24	Data
	25-32	Time in seconds
	33-42	Time in milliseconds
	43-47	Multiplex number.

The record on the output tape is in the following form:

1st word	Bits	6-17	Number of words in the block
	Bits	18-23	File number
	Bits	24-35	The number of the wave length scan
	Bits	36-47	Block number
2nd word	Bits	30-46	High order time from 1st tape (seconds of day)
3rd word	Bits	30-46	High order time from 2nd tape (should agree with 2nd word)
4th word on			Data in the packed-word concept

On the first record on the tape, two additional words are placed after the ones described above. These are the project codes derived from the ADC process input tapes. These are placed at the end of the record so that the first portion of all the records will be identical. Since a large number of output tapes will be generated, it is anticipated that a label equation card would be used to equate the output files (ZZZZZ01, ZZZZZ02, and ZZZZZ03) to some other identifier. A label equation card equates the file identifier which is used in the program to a file identifier which will be read off an input file or written on an output file.

Commutator Converter Program

This program converts the digital tape which is obtained from the commutated data into a form suitable for the Burroughs computer. It has a similar logic pattern as the FM Data Conversion Program and the output is essentially identical. The program generates the output tape and periodically lists the tape so that the conversion process can be checked.

The program has as its input:

(1) FILE FIL1, in free field format, the total number of blocks to be processed, the interval between the records which are to be listed, and the number of files on the input tape,

(2) FILE FCMU, tape unit, unlabeled tape, the digital tape containing the commutated data.

If the entire tape is to be processed, the number of blocks to be processed is a number in excess of the number of blocks on the tape. There are usually around 5000 blocks on a tape, so as input, the number of blocks to be processed is conveniently placed at 99999999.

The program has as its output:

(1) FILE FIL2, line printer, a listing of the data as they are converted at the interval specified by the input data on FIL1, and after the tape has been rewound, a listing, again at the proper interval, of the output tape (During the conversion process, when an end-of-file condition occurs on the input tape, the program writes the last block read of one file and the first block of the next file, but does not change the file identifier of the output file.),

(2) FILE FDTC1, labeled tape, a digital tape written in the packed-word concept, containing a single file with all the input data included.

The data are recorded onto FDTC1 in a packed-word which is of the following format:

Bits	0-13	Unused
	14-24	Data
	25-32	Time in seconds
	33-42	Time in milliseconds
	43-48	Multiplex number (commutator segment)

The first word in every record is the block number, and the second word contains the time of day in seconds in bits 30 through 46. The words from the third on to the end of the record contain the data in the packed-word as mentioned before. On the first record, however, the last word in the record is the project code.

Through a stream procedure, the number of words per record is automatically calculated by the computer. As a tape record is read into the buffer, a word count is generated in the first word prior to the start of the buffer. Thus, by referencing the buffer's starting address and going back one word, the number of words read can be made available to the computer. However, during the reading of the 36 bit word length tape, this is the number of 48 bit B-5500 words that are read and not the number of 36 bit words that are on the tape, thus, the number has to be converted before it is used.

The listing gives the time of day in hours, minutes, and seconds; time of day in seconds; number of words in the block to be listed; and the block number of output tape. If there are insufficient data to finish filling a row, zeroes are placed in the row to complete it.

A label equation card should be used if more than one tape is to be generated. Its use has already been described in the section of the FM Data Converter Program. In this case, the label equation card would equate the file identifier which appears in the program, FDTC1, to a file identifier which would be written on the output tape.

With the use of the label equation card, it is believed that changes to the program itself would not be necessary. Therefore, the program was processed by the RECC library SQUEEZE routine for easier handling of the card deck. This program removes all comments and all non-essential spaces from the program, and thereby reduces the number of cards on which the program is written.

Addition of Wave Length Program

This program is the final link between the data tapes as they are received from the analog-to-digital conversion and the main data reduction program. This program calculates wave length and adds it to the data from the FM Data Conversion Program. The resulting tape is then used as input to the Main FM Reduction Program. This program also periodically lists the output to provide a check on the calculations.

The input file for the program is FIL3, and the data should be arranged in the following order:

Card Number	Columns	
1	1-4	Number of input tapes
	10-14	Interval between listings of output
2	1-7	Wave length at first opening
	10-17	Wave length at second opening
	20-27	Wave length at third opening
	30-37	Wave length at fourth opening
	40-47	Wave length at fifth opening (if present)
3	1-4	File number for present tape and scan
	10-14	Scan number
4,5,6,7,8	14-23	Times for switch openings.

Cards 3 through 8 are then repeated for all the switch opening times that are present.

The scan number is simply a tally of the number of complete wave length scans for a given file, and is used to distinguish one scan from another without having to resort to time as the indicator. The scan number and file number are used by the program for "bookkeeping" so that it can maintain the correct correlation between the input cards and the input tape. The output tape is in the same format as the input tape except for the construction of the first word in each record. This word is changed to the following format:

Bits	6-17	Number of words in the record
	18-23	File number corresponding to input file
	24-35	Scan number for the particular file being processed
	36-47	Block number.

The output of this program is periodically listed on the printer file, FIL2, as determined by the second entry on card 1. This listing is in a similar format to that used in the previous listings that have been produced. This format facilitates cross checking between input and output and also the checking of the overall performance of the addition of wave length program.

It is recognized that the hand editing that is required prior to the running of this program is a weak link in the data reduction chain. Present plans attempt to overcome this weakness and to make the data reduction completely handled by computers, but this program does provide the necessary results. Plans call for a program that does the editing task and computes the wave length which is then added to the output tape from the FM data conversion program.

Calibration Program

This program performs the operations derived in detail in Appendix B. The program essentially performs four Fourier analyses: 1) expanding the photomultiplier tube signals in terms of the angles which are determined from the AC generator, 2) expanding the components determined in the first step in terms of the relative angle of incident polarization, 3) re-expanding the photomultiplier tube signal in terms of the true angle of the retardation plates, and 4) re-expanding the components in terms of the true angle of incident polarization. The first two expansions are necessary to determine the relative phase angles involved in the AC generator and the incident polarization. The program uses an input tape containing the raw calibration data as they are generated by the Addition of Wave Length Program.

Input parameters to the program are in the following format, right justified:

	Columns	Description
Card 1	1-4	Number of angles of incident polarization
	5-8	Number of sample points per angle of incident polarization
	9-12	The number of wave lengths at which calibration is desired
Card 2	1-6	Starting time for calibration procedure
	7-12	Digitized value to indicate a change of angle of incident polarization
	13-18	Wave length below which only the 541F tube is calibrated
	19-24	Wave length above which only the 541A tube is calibrated
Card 3	1-10	Interval to dump program
	11-20	Approximate frequency of AC generator
	21-30	Incremental change of angle of incident polarization
	31-40	η for the 541F tube (if known)
	41-50	η for the 541A tube (if known)
	51-60	ζ for the 541F tube (if known)
	61-70	ζ for the 541A tube (if known)

	Columns	Description
Card 4	1-2	Compute option for first AC generator approximation
	3-4	Compute option for second AC generator approximation
	5-6	Compute option for first pass of PM current data
	7-8	Compute option for first pass of component data
	9-10	Compute option for second pass of PM current data
	11-12	Compute option for second pass of component data
	21-25	TRUE if the input data is to be listed, FALSE otherwise
	26-30	TRUE if times at which the calibration wave lengths occurred are to be listed, FALSE otherwise
	31-35	TRUE if AC generator frequency iterations are to be listed, FALSE otherwise
	36-40	TRUE if the AC generator frequency with correction calculations is to be listed, FALSE otherwise
	41-45	TRUE if the true AC generator frequency is to be listed, FALSE otherwise
	46-50	TRUE if calibration parameters are to be punched also, FALSE otherwise
	51-55	TRUE if card images are to be listed as punched, FALSE otherwise (only valid when preceding entry is TRUE)
	56-60	TRUE if ζ and η are known for 541F tube, FALSE otherwise
	61-65	TRUE if ζ and η are known for 541A tube, FALSE otherwise
Card 5	6 columns	Wave lengths at which calibration is desired. The number of wave lengths should match the third entry on card 1.

The constants η and ζ are defined by equations (B-6) and (B-3) respectively.

The compute option pertains to the least squares curvefitting procedure that it utilized in the program. If the option variable is 1, then only the curvefitting portion of the procedure is used. However, if the option variable is different from 1, the following actions take place:

Option variable	Action
2	Standard deviation is calculated and printed.
3	Standard deviation is calculated and printed, data lying outside a specified number of standard deviations is discarded, the curve-fitting procedure is again entered and standard deviation is calculated using this new data.
4	Same as 2 except that the input data is listed.
5	Same as 3 except that both sets of input data to the curvefitting procedure are listed.

If enough previous runs of the program have given a good indication of the value of ζ and η for either or both PM tubes, these data can be introduced for subsequent runs to eliminate the calculation of these quantities. The elimination of these computations can decrease the computer time necessary for a complete run. Data processor and input-output times are calculated and printed as the last item in the output. Past experience indicates that approximately 300 seconds are required for each tube at a given wave length if ζ and η are not known.

Commutator Reduction Program

This program examines the commutated data, averages it over a suitable time interval, converts the raw digital value to engineering units such as degrees, volts, etc., and prints the output. Basically the program executes a disk sort to extract the various multiplexers and arrange their values in an array. A suitable calibration of the various channels must be made prior to the running of the program so that this information may be included in the program.

The basic problem in this program is that the data are arranged on the input tape in serial form, i.e., ascending multiplex number, but in order for it to be manipulated, it must be in a parallel form, i.e., all data of one multiplex number grouped together. The program utilizes a modified disk sort on the computer to accomplish this serial to parallel conversion. The program contains all calibrations of each segment including the nonlinear characteristic of the thermistors.

Some time prior to each flight a calibration was made on pertinent commutator channels. For those channels which measure a voltage, a known voltage was used as input, and the voltage out of the commutator segment was measured. A graph of these data was drawn relating voltage in versus voltage out. For the thermistor channels, the thermistors are subjected to different temperatures and their corresponding resistance is determined. Hence a relation between temperature and resistance is obtained. These data are contained in the program under a real procedure called RES(X). By determining the resistance, X, which is necessary to produce the voltage out of the commutator segment, and using this resistance as the input to the procedure, the procedure performs a tabular look up and produces a temperature which corresponds to the given resistance. For those commutator segments which measure an angle, a relation is derived between the angle and the voltage from the commutator segment, and this relation is used to obtain the desired angle.

In the program, double subscripted array B contains the data points for each segment in this manner: the first subscript is the segment or multiplex number (subscript 0 is the time), and the second subscript is the tally of the data points. The program reads the input tape, searches for segment #1, and fills array B[1, *] with its data point and fills array B[0, *] with the time. The data are then written on the disk. The program continues reading the input tape until 760 data points have been read into the array. Subsequently, the disk is rewound, read and searched for segment #2 and the array is filled until the data on the disk are exhausted. The disk is then rewound, read and searched for segment #3, the array is filled, and so on until all twenty-eight segments are read and placed into the array. When array B is completely filled, the majority of the work is finished. The program subsequently uses array B, converts to a voltage and averages the values over a suitable number as shown below:

Segments	Averaged over:
1,2,10,11,12,13 14,15,16,25,27	All 760 (approximately) samples
17,18,19,28	Every 150 samples
3,4,6,7,8,9,23,24	Every 75 samples
5	Every 25 samples

The averaging procedure uses all the data presented to it, calculates an average and the standard deviation. It then discards all data lying outside one standard deviation and reaverages the remaining data. The average voltage thus obtained is then applied to the particular calibration and the results of angle, voltage, or temperature are obtained.

Main FM Reduction Program

This program reduces the input data and obtains the intensity and the other Stokes parameters by performing a Fourier analysis of the photomultiplier tube currents in terms of the frequency of the AC generator and using the calibration constants that are also read into the program. An average wave length is calculated as well as the altitude and scatter angle. The output is listed for a given altitude and scatter angle (determined by the time) by intensity and the other modified Stokes parameters versus wave length. If the AC generator malfunctions, the program does not attempt the Fourier analysis but rather does a simple averaging and flags the data as being qualitative and not quantitative.

In brief, the program reads in and initializes all essential constants and calibration parameters. Secondly, the program positions the main input tape to the correct starting time and fills a stratch tape with the data to be used in subsequent calculations. The program then uses this scratch tape as input, fills the proper arrays with the photomultiplier currents, AC generator signals, and wave length. The AC generator signal is then used to calculate the angular position of the phase plates, and then a Fourier analysis is made on the photomultiplier tube signals. After the components are determined, they are used to calculate the Stokes vector of the incident light.

The program requires as card input on file FIL1 the following information in the correct format:

Card 1	Columns	1-5	Starting time
		11-15	Stopping time
		21-24	Wave length interval for the samples
Card 2	Columns	1-5	Minimum value for the "shutter"
		11-15	Maximum value for the "shutter"
		21-26	1st tube shutter attenuation
		31-36	2nd tube shutter attenuation
		41-46	Approximate generator frequency
Card 3	Columns	1-7	1st tube generator phase angle
		11-17	2nd tube generator phase angle
		21-24	Minimum wave length for 2nd tube
		26-29	Maximum wave length for 1st tube
		31-33	Compute option for generator data
		34-36	Compute option for PM data
		37-41	Print option for generator data
		42-46	Print option for PM data

The "shutter" is an auxiliary slit which automatically covers the entrance slit of the spectrometer when the instrument is pointing close to the sun. It effectively reduces the amount of light entering each photomultiplier by the amount of the third and fourth entries on card 2. The minimum wave length for the second tube (third entry, card 3) is that wave length below which only the short wave length tube is used to obtain data Similarly, the maximum wave length for the first tube is that wave length above which only data from the longer wave length tube is used. The compute option is the same as listed in the calibration program, but the print option (Boolean) determines if debugging and checking data is to be printed as well as the final results. If it is felt that the program is performing properly, then both print options should be set to FALSE, and the output would then consist of simply a listing of wave length versus the modified Stokes parameters.

```
COMMENT -- COMMUTATOR CONVERTER PROGRAM                                    ;
BEGIN INTEGER I,J,K,M,N,K1,K2,PROJCODE,BLOCK,P, IFIL, IFIL1,TS             ;
      REAL RNTIM, RNTIM1                                                   ;
      LABEL L1, L2, L3, L4, L0                                             ;
      SAVE ARRAY A[0:500]       ; ALPHA ARRAY B[0:600]                     ;
      ARRAY U[0:250]                                                       ;
STREAM PROCEDURE TST1(A,B); BEGIN SI←A;DI←B;SI←SI-8;DS←8 CHR END           ;
STREAM PROCEDURE STRM1(A,B,I,J,K);VALUE I,J,K;BEGIN SI←A;DI←B;          DI
←DI+2;DS←6 CHR;DI←DI+5;DS←3 CHR;SI←SI+3;I(J(DI←DI+2;DS←6 CHR));K(DI←DI+2
;DS←6 CHR) END;STREAM PROCEDURE STRM2(A,B,I,J,K);VALUE I,J,K;BEGIN SI←A;
DI←B;DI←DI+5;DS←3 CHR;SI←SI+3;I(J(DI←DI+2;DS←6 CHR));K(DI←DI+2;DS←6 CHR)
END;PROCEDURE OUT1(A,N,FILEID);VALUE N;INTEGER N;FILE FILEID;ALPHA ARRAY
 A[0];BEGIN INTEGER I,J,TM,THR,TMIN,TSEC;ARRAY B[ 1:12];FORMAT OUT FMT1(
I2, " HOURS", I3, " MINUTES",I3, " SECONDS",X10,I10,X45,"BLOCK NUMBER ",
I8),FMT2(4(F3.0,X1,F6.3,X1,F7.1,X9        ));FOR I←1   DO BEGIN TM←A[I]
.[30:17];THR←TM DIV 3600;TMIN←(TM-3600×THR) DIV 60;TSEC←TM-3600×THR-60×T
MIN;WRITE(FILEID,FMT1,THR,TMIN,TSEC,TM,A[0])END;FOR I←3 STEP 4 UNTIL N
 DO BEGIN FOR J←0 STEP 1 UNTIL 3 DO BEGIN
IF (I+J)>N THEN A[I+J]←0;
                                         B[3×J+1]←A[I+J].[43:5]×1.0;B[3
×J+2] ← A[I+J].[25:8]+A[I+J].[33:10]/1000;B[3×J+3]←A[I+J].[14:11]×1.0
                            END;WRITE(FILEID,FMT2,FOR J←1 STEP 1 UNTIL 12 D
O B[J]) END;END OUT1                                                       ;
      FORMAT OUT FMT0 ("RUN TIME ", F12.5, "SECONDS"),
                 FMT1 (X90, "PROJECT CODE ",I11),
                 FMT2(X56, "WORD COUNT ",I5)                               ;
      FILE IN FIL1 (1,10), FCMU (1, 200)                                   ;
      FILE OUT FIL2 1 (1,15)                                               ;
      FILE FDTC1 2 (1,800)                                                 ;
PROCEDURE DATE(FILEID)                                                     ;
      FILE FILEID                                                          ;
BEGIN ALPHA D ; INTEGER I, DAY, YR                                         ;
      INTEGER ARRAY MO[1:12]                                               ;
      FORMAT OUT FMT("RUN DATE ", I2,"/",I2,"/",I2)                        ;
      FILL MO[*] WITH 31, 28,31,30,31,30,31,31,30,31,30,31                 ;
      D ← TIME(0)                                                          ;
      YR ← 10×D.[18:6] + D.[24:6]                                          ;
```

```
      DAY ← 100×D.[30:6] + 10×D.[36:6] + D.[42:6]                      ;
      I ← 0                                                             ;
      FOR I ← I+1 WHILE (DAY > 0) DO                              BEGIN
           DAY ← DAY - MO[I]                                            ;
           IF ((YR MOD 4) = 0) THEN IF (I=2) THEN DAY←DAY-1       END   ;
      DAY ← DAY + MO[I]                                                 ;
      I ← I - 1                                                         ;
      WRITE(FILEID, FMT, I,DAY, YR)                         END   DATE  ;
      RNTIM ← TIME(2)                                                   ;
      DATE(FIL2)                                                        ;
      IFIL ← 0                                                          ;
      BLOCK ← 0                                                         ;
      READ (FIL1, /,    M, P, IFIL1)                                    ;
      CLOSE (FIL1, RELEASE)                                             ;
L0:   READ (FCMU[NO],    1, U[*])[L2:L0]                                ;
      TST1 (FCMU(0), TS)                                                ;
           N ← 4×((TS   ) DIV 3) + (TS   ) MOD 3                        ;
      WRITE (FIL2[NO], FMT2, N)                                         ;
      K1 ← (N-2) DIV 63                                                 ;
      K2 ← (N-2) MOD 63                                                 ;
      STRM1 (FCMU(0), A[0], K1,63,K2)                                   ;
      FOR I ← N-1 STEP -1 UNTIL 2 DO                              BEGIN
           B[I+1].[14:29] ← A[I].[12:29]                                ;
           B[I+1].[43:5] ← A[I].[43:5]                                  ;
           B[I+1].[12:2] ← 0                                      END   ;
      B[2] ← B[1] ← A[1]                                                ;
      B[N+1] ← A[0]                                                     ;
      B[0] ← BLOCK ← BLOCK + 1                                          ;
      WRITE (FDTC1, N+2, B[*])                                          ;
      OUT1(B, N   , FIL2)                                               ;
      RELEASE (FCMU)                                                    ;
L1:   READ (FCMU[NO], 1, U[*])[L2:L1]                                   ;
      TST1 (FCMU(0), TS)                                                ;
           N ← 4×(TS DIV 3) + TS MOD 3                                  ;
      K1 ← (N-1) DIV 63                                                 ;
      K2 ← (N-1) MOD 63                                                 ;
      STRM2 (FCMU(0), A[0], K1,63,K2)                                   ;
```

```
        FOR I ← N-1 STEP -1 UNTIL 1 DO                                        BEGIN
              B[I+2].[14:29] ← A[I].[12:29]                                        ;
              B[I+2].[43:5] ← A[I].[43:5]                                          ;
              B[I+2].[12:2] ← 0                                               END  ;
        BLOCK ←BLOCK + 1                                                           ;
        B[2] ← B[1] ← A[0]                                                         ;
        B[0] ← BLOCK                                                               ;
        WRITE (FDTC1, N+2, B[*])                                                   ;
        RELEASE (FCMU)                                                             ;
        IF (BLOCK MOD P) = 0                                                  THEN
BEGIN
        WRITE (FIL2[NO], FMT2, N)                                                  ;
              OUT1 (B, N+1, FIL2)                                                  ;
END ;
        IF (BLOCK ≤ M)                                                        THEN
              GO TO L1                                                             ;
L2:     IFIL ← IFIL + 1                                                            ;
        WRITE (FIL2[NO], FMT2, N)                                                  ;
        OUT1 (B, N+1, FIL2)                                                        ;
        IF IFIL < IFIL1 THEN GO TO L0                                              ;
        REWIND (FDTC1)                                                             ;
        LOCK (FCMU, RELEASE)                                                       ;
        RNTIM1 ← (TIME(2) - RNTIM)/60                                              ;
        WRITE (FIL2, FMT0, RNTIM1)                                                 ;
        READ (FDTC1[NO], 1, U[*])                                                  ;
        TST1 (FDTC1(0), N)                                                         ;
        READ (FDTC1,N   ,B[*])                                                     ;
        PROJCODE ← B[N-1]                                                          ;
        WRITE (FIL2[PAGE]) ;
        WRITE (FIL2, FMT1, PROJCODE)                                               ;
        SPACE (FDTC1, -1)                                                          ;
        WRITE (FIL2[NO], FMT2, N)                                                  ;
        N ← N - 1           ;
L3:     OUT1 (B, N-1, FIL2)                                                        ;
        SPACE (FDTC1, P-1)[L4]                                                     ;
        TST1(FDTC1(0), N)                                                          ;
        READ (FDTC1, N   , B[*])[L4]                                               ;
```

```
        WRITE (FIL2[NO], FMT2, N)                                              ;
        GO TO L3                                                               ;
L4:     LOCK (FDTC1, RELEASE)                                                  ;
        RNTIM ← (TIME(2) - RNTIM)/60                                           ;
        WRITE (FIL2, FMTO, RNTIM)                                              ;
END .
```

```
COMMENT -- FM DATA CONVERTER PROGRAM                                          ;
BEGIN INTEGER I,N,N1,N2,K1,K2,P,P1,TP,M,IFIL, IFL1,IF1,LBL,J1,J2,K3,K4,
IPAR                                                                          ;
      REAL RNTIM, RNTIM1, TM1, TM2, TM3, TM4                                  ;
      ARRAY CT1, CT2[0:6], TIM[1:2,0:3,0:350], VAL[0:8]                       ;
      INTEGER ARRAY IFL[0:6]                                                  ;
      SAVE ARRAY A,B[0:275]                                                   ;
      ALPHA ARRAY DT[0:1022]                                                  ;
      BOOLEAN Z                                                               ;
      LABEL LFIL,LEND,L1,L1A,L1B,L2,L2B,L3,L4,LP1,LP2,LP3,LP4,LP5,LP6,
            LA, LP1A, LP3A, LP5A, LP2A, LP4A, LP6A, L1C, L1D, L5              ;
      LABEL L8,L9,L10,L11                                                     ;
      SAVE FILE IN FFMD1 2 (1,250), FFMD2 2 (1,250), FFMD3 2 (1,250),
                   FFMD4 2 (1,250), FFMD5 2 (1,250), FFMD6 2 (1,250)          ;
      FILE OUT ZZZZZ01 2 (1,1023, SAVE 100),
               ZZZZZ02 2 (1,1010, SAVE 30),
               ZZZZZ03 2 (1,1010, SAVE 30)                                    ;
      FILE IN FIL7 (1,10)                                                     ;
      FILE OUT FIL2 6 (2,15)                                                  ;
      FILE OUT FILP1 2 (1,10, SAVE 1)                                         ;
      SWITCH FILE SWI1 := FFMD1, FFMD3, FFMD5                                 ;
      SWITCH FILE SWI2 := FFMD2, FFMD4, FFMD6                                 ;
      SWITCH FILE SWO1 := ZZZZZ01,ZZZZZ02,ZZZZZ03,ZZZZZ01,ZZZZZ01,
                          ZZZZZ01                                             ;
STREAM PROCEDURE STRM1(A,B,I,J,K);VALUE I,J,K; BEGIN SI:=A;DI:=B              ;
SI:=SI+2;DI:=DI+4;DS:=4 CHR;DI:=DI+5;DS:=3 CHR;SI:=SI+3                       ;
I(J(DI:=DI+2;DS:=6 CHR));K(DI:=DI+2;DS:=6 CHR)                    END         ;
STREAM PROCEDURE STRM2(A,B,I,J,K);VALUE I,J,K; BEGIN SI:=A;DI:=B              ;
                               DI:=DI+5;DS:=3 CHR;SI:=SI+3                    ;
I(J(DI:=DI+2;DS:=6 CHR));K(DI:=DI+2;DS:=6 CHR)                    END         ;
PROCEDURE DATE(FILE1) ;FILE FILE1 ;BEGIN ALPHA D;INTEGER I,DY,YR              ;
INTEGER ARRAY MO[1:12];FORMAT OUT F1("RUN DATE ", I2,"/",I2,"/",I2/)          ;
FILL MO[*] WITH 31,28,31,30,31,30,31,31,30,31,30,31; D:=TIME(0)               ;
YR:=10×D.[18:6]+D.[24:6];DY:=100×D.[30:6]+10×D.[36:6]+D.[42:6]-1              ;
I:=0;FOR I:=I+1 WHILE DY>0 DO BEGIN DY:=DY-MO[I];IF YR MOD 4=0 THEN
IF I=2 THEN DY:=DY-1 END;I:=I-1;DY:=DY+MO[I];WRITE(FILE1,F1,I,DY,YR)END;
      INTEGER PROCEDURE WORD(F1, LBL) ; FILE F1 ; INTEGER LBL                 ;
```

82

```
BEGIN LABEL L1,L2,L3  ;
      INTEGER I    ; ARRAY A[0:2]  ; STREAM PROCEDURE S1(A,B); BEGIN
SI:=A;DI:=B;SI:=SI-8;DS:=8 CHR END; READ(F1[NO],1,A[*])[L1:L2]         ;
      S1(F1(0),I) ; LBL := 0 ; WORD := I  ; GO TO L3  ;
      L1: LBL := 1 ; GO TO L3 ; L2: LBL := 2  ; L3:                END  ;
PROCEDURE OUT2(A,N,F1); VALUE N; INTEGER N; FILE F1; ALPHA ARRAY A[0]  ;
BEGIN INTEGER I,J,TM,T1,T2,T3;INTEGER ARRAY B[0:20];FORMAT OUT FMT1(I2,"
 HOURS",I3," MINUTES",I3," SECONDS",X10,"TIME ",I6,X44,"BLOCK NUMBER ",I
3,"-",I4),FMT2(4(I2,X2,I3,".",I3,X2,I4,".0",X7))                        ;
      J := A[0].[12:6]                                                  ;
FOR I:=1,2 DO BEGIN TM:=A[I].[30:17];T1:=TM DIV 3600;T2:=(TM-3600×T1)DIV
 60;T3:=TM-3600×T1-60×T2;WRITE(F1,FMT1,T1,T2,T3,TM,J,A[0].[18:30])END  ;
FOR I:=3STEP 4UNTIL N-3 DO BEGIN FOR J:=0,1,2,3 DO BEGIN IF I+J>N THEN A
[I+J]:=0;B[4×J]:=A[I+J].[43:5]                                          ;
         B[4×J+1]:=A[I+J].[25:8 ]                                       ;
         B[4×J+2]:=A[I+J].[33:10]                                       ;
         B[4×J+3]:=A[I+J].[14:11]                                 END   ;
         WRITE(F1,FMT2,FOR J:=0 STEP 1 UNTIL 15 DO B[J]) END END        ;
      FORMAT OUT FMTA("ACCUMULATED RUN TIME ", F8.2),
                 FMTB(X90, "PROJECT CODE ", I8),
                 FMTC(X57,"WORD COUNT ", I5),
                 FMTD("PARITY ERROR IN NEXT BLOCK AFTER ", I3,"-",I4,
                      "; TIMES: ", I5,", ",I5),
                 FP1("XXX PARITY ERROR, INPUT TAPE 1, FILE # ", I3),
                 FP2("XXX PARITY ERROR, INPUT TAPE 2, FILE # ", I3),
                 FP3("XXX PARITY ERROR, INPUT TAPE 1, STEP 2, FILE#",I3)
                ,FP4("XXX PARITY ERROR, INPUT TAPE 2, STEP 2, FILE#",I3)
                ,FP5("XXX PARITY ERROR, OUTPUT TAPE, STEP 1, FILE#",I3)
                ,FP6("XXX PARITY ERROR, OUTPUT TAPE, STEP 2, FILE#",I3);
PROCEDURE CALC1(N)                                                      ;
      VALUE N                                                           ;
      INTEGER N                                                         ;
BEGIN INTEGER I,I1,I2,J,K                                               ;
      REAL TM, T                                                        ;
OWN   BOOLEAN Z                                                         ;
      LABEL L1,L2                                                       ;
      OWN ARRAY B[0:1,0:10]                                             ;
```

```
      FORMAT OUT FMTP ("01", X2, I2, X2, I3, X2, F9.3, " GRATING",X50),
                 FMTR ("02", X2, I2, X2, I3, X2, F9.3, " CALIBRATION",
                                                 X46)                   ;
      TM ← A[I2].[30:17] - A[2].[23:8]                                  ;
      FOR I := 4 STEP 4 UNTIL N-1 DO                              BEGIN
           T ← A[I].[12:11]                                             ;
           J ← 8                                                        ;
           DO J ← J - 1 UNTIL T ≥ VAL[J]                                ;
           FOR I1 := 0,1 DO
                FOR K := 1 STEP 1 UNTIL 9 DO
                     B[I1,K-1] := B[I1,K]                               ;
           B[0,9] ← J                                                   ;
           B[1,9] ← TM + A[I].[23:8] + A[I].[31:10]/1000                ;
           FOR J := 1,3,5 DO                                      BEGIN
                FOR I1 := 0 STEP 1 UNTIL 4 DO
                     IF B[0,I1] ≠ J                          THEN
                            GO TO L1                                    ;
                FOR I1 := 5 STEP 1 UNTIL 9 DO
                     IF B[0,I1] ≠ J-1                        THEN
                            GO TO L1                                    ;
           CT1[IFIL] := CT1[IFIL] + 1                                   ;
           TIM[1,IFIL,CT1[IFIL]] := B[1,5]                              ;
      WRITE (FILP1, FMTP, IFIL, CT1[IFIL], TIM[1,IFIL,CT1[IFIL]])       ;
           Z := FALSE                                                   ;
L1:                                                               END   ;
      IF NOT Z                                               THEN BEGIN
           FOR I1 := 0 STEP 1 UNTIL 3 DO
                IF B[0,I1] ≥ 4                               THEN
                     GO TO L2                                           ;
           FOR I1 := 6 STEP 1 UNTIL 9 DO
                IF (B[0,I1] < 4) OR (B[0,I1] > 5)            THEN
                     GO TO L2                                           ;
           Z := TRUE                                                    ;
           CT2[IFIL] := CT2[IFIL] + 1                                   ;
           TIM[2,IFIL,CT2[IFIL]] := 3.0 + B[1,5]                        ;
      WRITE (FILP1, FMTR, IFIL, CT2[IFIL], TIM[2,IFIL,CT2[IFIL]]) END   ;
L2:                                                               END
```

```
        END                                                              ;
        RNTIM := TIME(2)                                                 ;
        DATE(FIL2)                                                       ;
        IPAR := 0                                                        ;
        READ (FIL7, /, FOR I:=0 STEP 1 UNTIL 7 DO VAL[I])                ;
        READ (FIL7, /, Z, P, P1, TP, M)                                  ;
        READ (FIL7, /, FOR I:=0 STEP 1 UNTIL TP-1 DO IFL[I])             ;
        IF1 := 0                                                         ;
        READ (FIL7, /, I)[L1B]                                           ;
        IF1 := 3                                                         ;
L1B:    CLOSE (FIL7, RELEASE)                                            ;
LP1A:
        DT[0] := 0                                                       ;
LFIL:   N1 := WORD (SWI1[IFIL], LBL)                                     ;
        IF LBL=1 THEN GO TO L8  ELSE IF LBL=2 THEN GO TO LP1             ;
        DT[0].[12:6] ← IFL1 + 1                                          ;
        N1 := 4×(N1 DIV 3) + N1 MOD 3                                    ;
        K1 := (N1-2) DIV 63                                              ;
        K2 := (N1-2) MOD 63                                              ;
        STRM1 (SWI1[IFIL](0), A[0], K1, 63, K2)                          ;
        CALC1(N1)                                                        ;
        RELEASE (SWI1[IFIL])                                             ;
LA:     N2 := WORD (SWI2[IFIL], LBL)                                     ;
        IF LBL=1 THEN GO TO L9  ELSE IF LBL=2 THEN GO TO LP2             ;
        N2 := 4×(N2 DIV 3) + N2 MOD 3                                    ;
        K1 := (N2-2) DIV 63                                              ;
        K2 := (N2-2) MOD 63                                              ;
        STRM1 (SWI2[IFIL](0), B[0], K1, 63, K2)                          ;
        RELEASE (SWI2[IFIL])                                             ;
        K3 := A[1]                                                       ;
        K4 := B[1]                                                       ;
        DT[1003] := A[0]                                                 ;
        DT[1004] := B[0]                                                 ;
        J1 := J2 := 2                                                    ;
LP3A:
L1:     N ← IF DT[0].[18:30] = 0 THEN 5 ELSE 3                           ;
        FOR I := 0 STEP 2 UNTIL 498 DO                             BEGIN
```

85

```
L1C:     IF I+J1+2 > N1                                        THEN  BEGIN
             J1 := 1-I                                                        ;
             N1 := WORD (SWI1[IFIL], LBL)                                     ;
         IF LBL=1 THEN GO TO L10 ELSE IF LBL=2 THEN GO TO LP3                 ;
             N1 := 4×(N1 DIV 3) + N1 MOD 3                                    ;
             K1 := (N1-1) DIV 63                                              ;
             K2 := (N1-1) MOD 63                                              ;
             STRM2 (SWI1[IFIL](0), A[0], K1, 63, K2)                          ;
             TM3 ← A[0].[30:17] - A[1].[23:8]                                 ;
             CALC1(N1)                                                        ;
             RELEASE (SWI1[IFIL])                                    END      ;
L1A:     IF I+J2+2 > N2                                        THEN  BEGIN
             J2 := 1-I                                                        ;
             N2 := WORD (SWI2[IFIL], LBL)                                     ;
         IF LBL=1 THEN GO TO L11 ELSE IF LBL=2 THEN GO TO LP4                 ;
             N2 := 4×(N2 DIV 3) + N2 MOD 3                                    ;
             K1 := (N2-1) DIV 63                                              ;
             K2 := (N2-1) MOD 63                                              ;
             STRM2 (SWI2[IFIL](0), B[0], K1, 63, K2)                          ;
             TM4 ← B[0].[30:17] - B[1].[23:8]                                 ;
             RELEASE (SWI2[IFIL])                                    END      ;
         IF A[I+J1] =0                                         THEN  BEGIN
             TM1 ← TM1 + 256.0                                                ;
             J1 := J1 + 1                                                     ;
             GO TO L1C                                               END      ;
         IF B[I+J2] = 0                                        THEN  BEGIN
             TM2 ← TM2 + 256.0                                                ;
             J2 := J2 + 1                                                     ;
             GO TO L1A                                               END      ;
         IF I = 0                                              THEN
             IF DT[0].[18:30] = 0                        THEN  BEGIN
                     K3 ← A[1].[30:17]-A[2].[23:8]                            ;
                     K4 ← B[1].[30:17]-B[2].[23:8]                   END
                                                               ELSE  BEGIN
                     K3 ← A[0].[30:17]-A[1].[23:8]                            ;
                     K4 ← B[0].[30:17]-B[1].[23:8]                   END      ;
         DT[2×I+3].[14:29] := A[I+J1  ].[12:29]                               ;
```

```
          DT[2×I+3],[43:5]   := A[I+J1   ],[43:5]                                      ;
          DT[2×I+4],[14:29]  := A[I+J1+1],[12:29]                                      ;
          DT[2×I+4],[43:5]   := A[I+J1+1],[43:5]                                       ;
          DT[2×I+5],[14:29]  := B[I+J2   ],[12:29]                                     ;
          DT[2×I+5],[43:5]   := 3                                                      ;
          DT[2×I+6],[14:29]  := B[I+J2+1],[12:29]                                      ;
          DT[2×I+6],[43:5]   := 4                                                      ;
      TM1 ← DT[2×I+3].[25:8] + DT[2×I+3].[33:10]/1000 + TM3                            ;
      TM2 ← DT[2×I+5].[25:8] + DT[2×I+5].[33:10]/1000 + TM4                            ;
      IF TM1 ≠ TM2                                     THEN          BEGIN
          IF ABS(TM1-TM2) = 0.001                      THEN
              GO TO L5                                                                 ;
          IF ABS(TM1-TM2) ≤ 0.700                      THEN   BEGIN
              IF TM1 > TM2                             THEN   BEGIN
                  J2 ← J2 + 2                                                          ;
                  GO TO L1A                                   END
                                                       ELSE   BEGIN
                  J1 ← J1 + 2                                                          ;
                  GO TO L1C                                   END                      ;
                                                              END                      ;
          DT[1].[30:17] ← K3 + DT[3].[25:8]                                            ;
          DT[2].[30:17] ← K4 + DT[5].[25:8]                                            ;
          WRITE (FIL2[N0], FMTC, N)                                                    ;
          WRITE (SW01[IFIL+IF1], N, DT[*])                                             ;
          OUT2 (DT, N+3, FIL2)                                                         ;
          IF TM1 < TM2                                 THEN   BEGIN
              N1 ← 0                                                                   ;
              J1 ← J2 ← 1                                                              ;
              GO TO L1                                        END                      ;
              N2 ← 0                                                                   ;
          J1 ← J2 ← 1                                                                  ;
          GO TO L1                                                           END       ;
          IF DT[2×I+3].[33:10] MOD 500=0 AND I≠498     THEN   BEGIN
              DT[0].[18:30] ← DT[0].[18:30] + 1                                        ;
              IF DT[0].[18:30] = 1                     THEN
                  OUT2 (DT, N-3, FIL2)                                                 ;
              WRITE (SW01[IFIL+IF1], N, DT[*])                                         ;
```

```
                J1 ← I + J1 + 2                                                   ;
                J2 ← I + J2 +2                                                    ;
                GO TO L1                                          END             ;
           N ← N +4                                                               ;
L5:                                                                     END       ;
       DT[0].[18:30] ← DT[0].[18:30] + 1                                          ;
       DT[1].[30:17] ← K3 + DT[3].[25:8]                                          ;
       DT[2].[30:17] ← K4 + DT[5].[25:8]                                          ;
       IF DT[0].[18:30] = 1                                   THEN
           OUT2 (DT, N-3, FIL2)                               ELSE
           FOR I ← 0 STEP 1 UNTIL P1-1 DO
                IF DT[0].[18:30] MOD P = I                    THEN
                     OUT2 (DT, N-1, FIL2)                                         ;
       WRITE (SWO1[IFIL+IF1], N, DT[*])                                           ;
       J1 := 500 + J1                                                             ;
       J2 := 500 + J2                                                             ;
       IF Z OR (DT[0].[18:30] ≤ M)                            THEN
           GO TO L1                                                               ;
L2:    IFL1 := IFL1 + 1                                                           ;
       OUT2 (DT, 1002, FIL2)                                                      ;
       DT[0] ← 0                                                                  ;
       IF IFL1 < IFL[IFIL]                                    THEN      BEGIN
           WRITE (FIL2[PAGE])                                                     ;
           GO TO LFIL                                                   END       ;
       IFL1 ← 0                                                                   ;
       LOCK (SWI1[IFIL], RELEASE)                                                 ;
       LOCK (SWI2[IFIL], RELEASE)                                                 ;
       IF IF1 = 0                                             THEN
           CLOSE (SWO1[IFIL+IF1], *)                          ELSE
           REWIND (SWO1[IFIL+IF1])                                                ;
       WRITE (FIL2[PAGE])                                                         ;
       IFIL := IFIL + 1                                                           ;
       IF IFIL < TP                                           THEN
           GO TO LFIL                                                             ;
       RNTIM1 := (TIME(2) - RNTIM)/60                                             ;
       WRITE (FIL2, FMTA, RNTIM1)                                                 ;
       WRITE (FIL2[PAGE])                                                         ;
```

```
        REWIND (FILP1)                                                   ;
        REWIND (SW01[IFIL+IF1-1])                                        ;
        IFIL := 0                                                        ;
LP5A:
L2B:    N := WORD(SW01[IFIL], LBL)                                       ;
        IF LBL = 1 THEN GO TO L4    ELSE IF LBL = 2 THEN GO TO LP4       ;
        READ (SW01[IFIL], N, DT[*])                                      ;
        WRITE (FIL2, FMTB, FOR I:=2,100 DT[N-I])                         ;
        WRITE (FIL2[NO], FMTC, N)                                        ;
        OUT2 (DT, N-3, FIL2)                                             ;
        SPACE (SW01[IFIL], P-2)[L4]                                      ;
L3:     FOR I := 0 STEP 1 UNTIL P1-1 DO                             BEGIN
LP6A:
            J1 ← DT[0].[12:6]                                            ;
            N := WORD (SW01[IFIL], LBL)                                  ;
            IF LBL = 1 THEN GO TO L4 ELSE IF LBL = 2 THEN GO TO LP6      ;
            READ (SW01[IFIL], N, DT[*])                                  ;
            IF DT[0].[12:6] ≠ J1                        THEN  BEGIN
                SPACE (SW01[IFIL], -DT[0].[18:30])                       ;
                GO TO L2B                                     END        ;
            IF (DT[0].[18:30] MOD P) ≠ I                THEN  BEGIN
                SPACE(SW01[IFIL],P+I-(DT[0].[18:30] MOD P))[L4:LP6]      ;
                OUT2 (DT, N-1, FIL2)                                     ;
                GO TO LP6A                                    END        ;
            WRITE (FIL2[NO], FMTC, N)                                    ;
            OUT2 (DT, N-1, FIL2)                                    END  ;
        SPACE (SW01[IFIL], P-P1)[L4]                                     ;
        GO TO L3                                                         ;
L4:     RNTIM1 := (TIME(2) - RNTIM)/60                                   ;
        WRITE (FIL2[PAGE], FMTA, RNTIM1)                                 ;
        IFIL := IFIL + 1                                                 ;
        IF IF1 = 0                                      THEN
        IF IFIL < TP                                    THEN
            GO TO L2B                                                    ;
        LOCK (SW01[IFIL], RELEASE)                                       ;
        GO TO LEND                                                       ;
L9:     SPACE (SWI1[IFIL], 1)[L2:L9]                                     ;
```

```
L8:    SPACE (SWI2[IFIL], 1)[L2:L8]                                              ;
L10:   SPACE (SWI2[IFIL], 1)[L2:L10]                                             ;
L11:   SPACE (SWI1[IFIL], 1)[L2:L11]                                             ;
LP1:   WRITE (FIL2[DBL], FP1, IFIL)                                              ;
       RELEASE (SWI1[IFIL])                                                      ;
       N1 := N2 := 0                                                             ;
       WRITE (FILP1, FMTD, DT[0].[12:6], DT[0].[18:30],DT[1].[30:17],
                               DT[2].[30:17])                                    ;
       IPAR := IPAR + 1                                                          ;
       IF IPAR > 10 THEN GO TO LEND                                              ;
       GO TO LP1A                                                                ;
LP2:   WRITE (FIL2[DBL], FP2, IFIL)                                              ;
       RELEASE (SWI2[IFIL])                                                      ;
       N1 := N2 := 0                                                             ;
       WRITE (FILP1, FMTD, DT[0].[12:6], DT[0].[18:30],DT[1].[30:17],
                               DT[2].[30:17])                                    ;
       IPAR := IPAR + 1                                                          ;
       IF IPAR > 10 THEN GO TO LEND                                              ;
       GO TO LP1A                                                                ;
LP3:   WRITE (FIL2[DBL], FP3, IFIL)                                              ;
       RELEASE (SWI1[IFIL])                                                      ;
       N1 := N2 := 0                                                             ;
       J1 := J2 := 1                                                             ;
       WRITE (FILP1, FMTD, DT[0].[12:6], DT[0].[18:30],DT[1].[30:17],
                               DT[2].[30:17])                                    ;
       IPAR := IPAR + 1                                                          ;
       IF IPAR > 10 THEN GO TO LEND                                              ;
       GO TO LP3A                                                                ;
LP4:   WRITE (FIL2[DBL], FP4, IFIL)                                              ;
       RELEASE (SWI2[IFIL])                                                      ;
       N1 := N2 := 0                                                             ;
       J1 := J2 := 1                                                             ;
       WRITE (FILP1, FMTD, DT[0].[12:6], DT[0].[18:30],DT[1].[30:17],
                               DT[2].[30:17])                                    ;
       IPAR := IPAR + 1                                                          ;
       IF IPAR > 10 THEN GO TO LEND                                              ;
       GO TO LP3A                                                                ;
```

```
LP5:   WRITE (FIL2[DBL], FP5, IFIL)                                ;
       RELEASE (SWO1[IFIL])                                        ;
       IPAR := IPAR + 1                                            ;
       IF IPAR > 10 THEN GO TO LEND                                ;
       GO TO LP5A                                                  ;
LP6:   WRITE (FIL2[DBL], FP6, IFIL)                                ;
       RELEASE (SWO1[IFIL])                                        ;
       IPAR := IPAR + 1                                            ;
       IF IPAR > 10 THEN GO TO LEND                                ;
       GO TO LP6A                                                  ;
LEND:
BEGIN  LABEL L1,L2                                                 ;
       FILE FILPN 0 (1,10)                                         ;
       ALPHA ARRAY A[0:9]                                          ;
       REWIND (FILP1)                                              ;
       WRITE (FIL2[PAGE])                                          ;
L1:    READ (FILP1, 10, A[*])[L2]                                  ;
       WRITE (FILPN, 10, A[*])                                     ;
       WRITE (FIL2,  10, A[*])                                     ;
       GO TO L1                                                    ;
L2:    CLOSE (FILP1, PURGE)                                        ;
END                                                                ;
END .
```

```
COMMENT -- FM DATA CONVERTER PROGRAM, PART 2, ADDITION OF WAVELENGTH    ;
BEGIN INTEGER I,J,L,N,SN,FL,TPI,TPO,BLK,INTP,LBLK                      ;
      REAL TM, LAMBDA                                                  ;
      ARRAY T[0:10,0:50,0:4], LMB[0:4], STT,STP[0:25]                  ;
      ALPHA ARRAY A[0:1022]                                            ;
      INTEGER ARRAY SCAN[0:40]                                         ;
      LABEL L1,L2,L3,L4,L5                                             ;
      BOOLEAN Z                                                        ;
      FILE IN  FIL3 (5,10),
               TSI06 (1,1010),
               TSI07 (1,1010),
               TSI08 (1,1010),
               TSI09 (1,1010),
               TSI10 (1,1010)                                          ;
      FILE     FMDATA6 2 "FMDATA" "FILE6" (1, 1010, SAVE 365),
               FMDATA7 2 "FMDATA" "FILE7" (1, 1010, SAVE 365),
               FMDATA8 2 "FMDATA" "FILE8" (1, 1010, SAVE 365),
               FMDATA9 2 "FMDATA" "FILE9" (1, 1010, SAVE 365)          ;
      SWITCH FILE SWIN ← TSI06, TSI07, TSI08, TSI09, TSI10             ;
      SWITCH FILE SWOUT ← FMDATA6, FMDATA7, FMDATA8, FMDATA9           ;
      FILE OUT FIL2 6 (5,15)                                           ;
      FORMAT OUT FMT1 (2(I8, X8), I5),
                 FMT2 (4(I4,".0",X1,I2,X1,I3,".",I3,X1,I4,".0",X3),I4),
                FMT3 (I2," HOURS",I3," MINUTES",I3," SECONDS", X10,
                      "TIME ", I6)                                     ;
FORMAT OUT FMTA (4(I10,X10))                                           ;
      PROCEDURE OUT3 (A, N, F1)                                        ;
           VALUE N                                                     ;
           INTEGER N                                                   ;
           ARRAY A[0]                                                  ;
           FILE F1                                                     ;
BEGIN INTEGER I,J,TM,T1,T2,T3                                          ;
      LIST LIST1 (T1,T2,T3,TM),
           LIST2 (FOR I ← 3 STEP 4 UNTIL N-3 DO [FOR J ← 0,1,2,3 DO [
    A[I+J].[1:13],A[I+J].[43:5], A[I+J].[25:8], A[I+J].[33:10], A[I+J].
                 [14:11]], (I+1) DIV 4])                               ;
      FOR I ← 1,2 DO                                              BEGIN
```

```
            TM ← A[I].[30:17]                                                    ;
            T1 ← TM DIV 3600                                                     ;
            T2 ← (TM - 3600×T1) DIV 60                                           ;
            T3 ← TM - 3600×T1 - 60×T2                                            ;
            WRITE (F1, FMT3, LIST1)                                       END    ;
        IF (N-3) MOD 4 ≠ 0                                  THEN
            FOR I ← N+1 STEP 1 WHILE (I+N-3) MOD 4 ≠ 0 DO
                A[I] ← 0                                                         ;
        WRITE (F1, FMT2, LIST2)                                                  ;
END                                                                              ;
        PROCEDURE WORD (FILEID, N, LEOF, LPAR)                                   ;
            FILE FILEID                                                          ;
            INTEGER N                                                            ;
            LABEL LEOF, LPAR                                                     ;
BEGIN   STREAM PROCEDURE S (A,B)                                                 ;
        BEGIN       SI ← A                                                       ;
                    DI ← B                                                       ;
                    SI ← SI - 8                                                  ;
                    DS ← WDS                                                     ;
        END                                                                      ;
        READ (FILEID[NO])[LEOF:LPAR]                                             ;
        S (FILEID(0), N)                                                         ;
END                                                                              ;
BEGIN   INTEGER I,J,K                                                            ;
        LABEL L1,L2                                                              ;
        FORMAT IN FMT1 (5(F7.2,X3)),
                  FMT2 (2(I4, X6)),
                  FMT3 (X13, F9.3)                                               ;
        LIST LIST1 (FOR I ← 0,1,2,3,4 DO LMB[I]),
            LIST2 (FOR K ← 0,1,2,3,4 DO T[I,J,K])                                ;
        READ (FIL3, FMT2, INTP, LBLK)                                            ;
        READ (FIL3, FMT1, LIST1)                                                 ;
L1:     READ (FIL3, FMT2, I, J)[L2]                                              ;
        READ (FIL3, FMT3, LIST2)                                                 ;
        GO TO L1                                                                 ;
L2:     CLOSE (FIL3, RELEASE)                                                    ;
END                                                                              ;
```

```
      FL ← -1                                                      ;
      TPI ← TPO ← SN ← 0                                           ;
L1:   WORD (SWIN[TPI], N, L3, L1)                                  ;
      READ (SWIN[TPI], N, A[*])                                    ;
      BLK ← BLK + 1                                                ;
      IF A[0].[18:30] < 10 AND NOT Z               THEN       BEGIN
           IF FL ≠ -1                              THEN
                SCAN[FL] ← SN                                      ;
           SN ← L ← 0                                              ;
           FL ← FL + 1                                             ;
           IF FL MOD 5 = 0  AND FL ≠ 0             THEN       BEGIN
           CLOSE (SWOUT[TPO], *)                                   ;
           TPO ← TPO + 1                                      END  ;
           STT[FL] ← A[1].[30:17]                                  ;
           BLK ← 1                                                 ;
           Z ← TRUE                                           END
                                                   ELSE
           IF A[0].[18:30] > 100                   THEN
                Z ← FALSE                                          ;
      STP[FL] ← A[1].[30:17]                                       ;
      A[0].[6:12] ← N                                              ;
      A[0].[18:6] ← FL + 1                                         ;
      A[0].[24:12] ← SN + 1                                        ;
      A[0].[36:12] ← BLK                                           ;
      IF A[2].[30:17] < T[FL,SN,0] - 1.0           THEN       BEGIN
           WRITE (SWOUT[TPO], N, A[*])                             ;
           GO TO L1                                           END  ;
      FOR I ← 4 STEP 4 UNTIL (IF BLK=1 THEN N-3 ELSE N-1) DO  BEGIN
           TM ← A[2].[30:17] - A[3].[25:8] + A[I].[25:8] + A[I].[33:10]/
                         1000                                      ;
           IF TM < T[FL,SN,0]                      THEN
                GO TO L2                                           ;
           IF T[FL,SN,L] ≠ T[FL,SN,L+1]            THEN
           LAMBDA ← LMB[L] + (TM-T[FL,SN,L])×(LMB[L+1]-LMB[L])/
                    (T[FL,SN,L+1]-T[FL,SN,L])                      ;
           FOR J ← -1,0,1,2 DO
                A[I+J].[1:13] ← ENTIER (LAMBDA+0.5)                ;
```

```
          IF TM > T[FL,SN,L+1]                                THEN  BEGIN
               L ← L + 1                                                    ;
               IF L ≥ 4                                       THEN  BEGIN
                    IF T[FL,SN+1,0] = 99999.999               THEN  BEGIN
                         WHILE A[0].[18:30]>10 DO
                              READ (SWIN[TPI], 1, A[*])[L3:L1]              ;
                         SPACE (SWIN[TPI], -1)                              ;
                         GO TO L1                                   END     ;
                    L ← 0                                                   ;
                    SN ← SN + 1                                     END     ;
                                                                    END     ;
L2:                                                                      END ;
     IF BLK MOD LBLK = 0                                      THEN
          OUT3 (A, N, F,L2)                                                 ;
     WRITE (SWOUT[TPO], N, A[*])                                            ;
     GO TO L1                                                               ;
L3:  SCAN[FL] ← SN                                                          ;
     A[0] ← 0                                                               ;
     REWIND (SWIN[TPI])                                                     ;
     TPI ← TPI + 1                                                          ;
     IF TPI < INTP                                            THEN
          GO TO L1                                                          ;
     WRITE (FIL2[PAGE])                                                     ;
     FOR I ← 0 STEP 1 UNTIL FL DO
          WRITE (FIL2, FMT1, STT[I], STP[I], SCAN[I])                       ;
END .
```

```
COMMENT -- COMMUTATOR REDUCTION PROGRAM (NEW)                                    ;
BEGIN INTEGER I,J,K,L,N,FIL,CNT1,CNT2,PAGE1,PAGE2,BLK                            ;
      INTEGER ARRAY M[0:8]                                                       ;
      REAL T1,T2,T3,SLOPE,V0,V1,V2                                               ;
      ALPHA ARRAY A[0:350]                                                       ;
      LABEL L1,L2,L3,L4                                                          ;
      LABEL LP1 ;
      ARRAY B[0:28,0:950], C[0:28,0:40], D[0:150], P,T[0:30]                     ;
      FILE OUT  FIL3 6 (5,15),
                FIL4 6 (5,15)                                                    ;
      FILE IN   FDTC1 "COMMUTR" "FDTC1" (1,300),
                FDTC2 "COMMUTR" "FDTC2" (1,300),
                FDTC3 "COMMUTR" "FDTC3" (1,300)                                  ;
      SWITCH FILE SWFIL ← FDTC1, FDTC2, FDTC3                                    ;
      FILE FDISK DISK SERIAL [1:800] (2,200)                                     ;
FORMAT OUT FMT1 (X43,"COMMUTATED DATA"/X40,"NATURAL SKY BACKGROUND"///,
                 X13,"ANGLE IN DEGREES     SOL",X6,"PC  CLUTCH VOLTAGES",
                 X12,"TEMPERATURE IN DEGREES F"/" TIME GMT     REL  SUN",
              "   SPT  AZ  MON   AZIMUTH DIFF ELEVATION DIFF             ",
                 "    PC  FRM  MNB  TMB  ELB"//),
           FMT1A(X43,"COMMUTATED DATA"/X40,"NATURAL SKY BACKGROUND"////,
                 X27,"VOLTAGES"/" TIME GMT      MAIN  ELCT   TM    REF",
                 "    THERM   PC"//),
           FMT2 (X113, "PAGE", I2),
           FMT3 (X68, X10, 5(I3,X2)),
           FMT4 (X27, I3, X3, F3.1, X2, 2(2(F4.1, X1), I2, X3)),
           FMT5 (I2,":",I2,":",F4.1, X2, 3(I4, X1)),
           FMT6 (I2,":",I2,":",F4.1, X30, 2(F4.1, X2)),
           FMT7 (X14, 4(F4.1, X2)),
           FMT8 (X5,"NUMBER OF BLOCKS USED",I10)                                 ;
LIST LIST1 (T1, T2, T3, P[1], P[3], P[2]),
      LIST2 (FOR L ← 4,5,6,8 DO P[L], T1, P[7], P[9], T2),
      LIST3 (FOR L ← 12 STEP 1 UNTIL 16 DO P[L]),
      LIST4 (P[18], P[19], P[17], P[28]),
      LIST5 (T1,T2,T3,V1,V2)                                                     ;
DEFINE FORJ = FOR J ← 0 STEP 1 UNTIL #                                           ;
REAL PROCEDURE MIN(A,B);VALUE A,B;REAL A,B;MIN←IF A<B THEN A ELSE B              ;
```

```
      PROCEDURE REED (F, A, N, LEOF)                              ;
           FILE F                                                  ;
           ARRAY A[0]                                              ;
           INTEGER N                                               ;
      LABEL LEOF                                                   ;
BEGIN STREAM PROCEDURE S (A,B)                                     ;
BEGIN SI ← A                                                       ;
      DI ← B                                                       ;
      SI ← SI - 8                                                  ;
      DS ← WDS                                                     ;
END S                                                              ;
      LABEL L1,L2,L3                                               ;
L1:   READ (F[NO])[LEOF:L2]                                        ;
      S(F(0), N)                                                   ;
      READ (F, N, A[*])                                            ;
      GO TO L3                                                     ;
L2:   RELEASE (F)                                                  ;
      GO TO L1                                                     ;
L3:
END REED                                                           ;
REAL  PROCEDURE AVG (A,N)                                          ;
      VALUE N                                                      ;
      ARRAY A[0]                                                   ;
      INTEGER N                                                    ;
BEGIN INTEGER I,M                                                  ;
      REAL T1,T2,AV,S                                              ;
      T1 ← T2 ← 0                                                  ;
      FOR I ← 0 STEP 1 UNTIL N-1 DO                          BEGIN
           T1 ← T1 + A[I]                                          ;
           T2 ← T2 + A[I]*2                                  END   ;
      AV ← T1/N                                                    ;
      S ← SQRT(T2/N - AV*2)                                        ;
      T1 ← T2 ← M ← 0                                              ;
      FOR I ← 0 STEP 1 UNTIL N-1 DO
           IF ABS(A[I]-AV) ≤ S                    THEN       BEGIN
                M ← M + 1                                          ;
                T1 ← T1 + A[I]                                     ;
```

```
             T2 ← T2 + A[I]*2                                        END  ;
    AVG ← T1/M                                                            ;
END AVG                                                                   ;
REAL PROCEDURE RES(X)          ; VALUE X            ; REAL X              ;
BEGIN INTEGER I                ; REAL T             ; ARRAY R,TP[0:32]    ;
    FILL R[*] WITH  35.17, 42.11, 50.09, 61.14, 76.35, 94.82, 120.90,
              136.77, 156.06, 176.71, 202.11, 230.23, 261.22, 297.62,
              337.00, 374.42, 437.58, 469.81, 489.87, 514.29, 561.43,
              589.82, 635.10, 670.49, 730.00, 790.00, 855.00, 926.00,
              1000.0, 1080.0, 1160.0                                      ;
    FILL TP[*] WITH 130.0, 120.0, 110.0, 100.0,  90.0,  80.0,   70.0,
                65.0,   60.0,   55.0,   50.0,   45.0,   40.0,   35.0,
                30.0,   25.0,   20.5,   18.0,   16.5,   15.0,   12.5,
              11.0,    9.0,     7.5,     5.0,     2.5,     0.0,   -2.5,
              -5.0, -7.5, -10.0                                           ;
    I ← -1                                                                ;
    DO I ← I + 1 UNTIL (R[I] > X) OR (I > 31)                             ;
    IF I > 31                                            THEN
        T ← 99999999.9                                   ELSE
        IF I = 0                                         THEN
            T ← (R[0] - X)×(TP[1] - TP[0])/(R[0] - R[1]) + TP[0]
                                                         ELSE
            T ← (R[I] - X)×(TP[I] - TP[I-1])/(R[I] - R[I-1])
                       +TP[I]                                             ;
    RES ← T                                                   END   RES   ;
        FIL ← 2                                                           ;
    SPACE (SWFIL[FIL], 1)                                                 ;
    CNT1 ← CNT2 ← BLK ← 0                                                 ;
    PAGE1 ← PAGE2 ← 1                                                     ;
    WRITE (FIL3, FMT2, PAGE1)                                             ;
    WRITE (FIL4, FMT2, PAGE2)                                             ;
    WRITE (FIL3, FMT1)                                                    ;
    WRITE (FIL4, FMT1A)                                                   ;
WHILE TRUE DO BEGIN
IF BLK ≥ 500 THEN GO TO L4 ;
    FOR I ← 0 STEP 1 UNTIL 28 DO
        M[I] ← -1                                                         ;
```

```
L1:   REED (SWFIL[FIL], A, N, L4)                                          ;
      BLK ← BLK + 1                                                        ;
      A[0] ← N                                                             ;
      FOR I ← 3 STEP 1 UNTIL N-1 DO
            IF A[I].[43:5] = 1                        THEN          BEGIN
                  B[1,M[1]←M[1]+1] ← A[I].[14:11]                          ;
                  B[0,M[0]←M[0]+1] ← A[1].[30:17]-A[3].[25:8]+A[I].[25:8]
                                    +A[I].[33:10]/1000              END    ;
      WRITE (FDISK, N, A[*])                                               ;
      IF M[1] ≤ 760                                   THEN
            GO TO L1                                                       ;
      REWIND (FDISK)                                                       ;
      FOR J ← 2 STEP 1 UNTIL 25, 27, 28 DO                          BEGIN
L2:         READ (FDISK[NO], 1, A[*])[L3]                                  ;
            N ← A[0]                                                       ;
            READ (FDISK, N, A[*])                                          ;
            FOR I ← 3 STEP 1 UNTIL N-1 DO
                  IF A[I].[43:5] = J                  THEN
                        B[J,M[J]←M[J]+1] ← A[I].[14:11]                    ;
            GO TO L2                                                       ;
L3:   REWIND (FDISK)                                                END    ;
      FOR I ← 1,2,25,27 DO
            D[I] ← AVG(B[I,*], M[I])                                       ;
      SLOPE ← 4.0/(D[1]-D[2])                                              ;
      V0 ← D[2] - 1.0/SLOPE                                                ;
      V1 ← (D[27] - V0)×SLOPE×3.84                                         ;
      V2 ← (D[25] - V0)×SLOPE×3.93                                         ;
      FOR I ← 0 STEP 25 UNTIL MIN(M[5],M[24]) DO                    BEGIN
            K ← I DIV 25                                                   ;
            FORJ 24 DO D[J] ← (B[5,I+J] - V0)×SLOPE                        ;
            FORJ 24 DO D[J+25] ← (B[24,I+J] - V0)×SLOPE                    ;
            C[5,K] ← AVG(D, 50)                                     END    ;
      FOR I ← 0 STEP 75 UNTIL MIN(M[3], M[10]) DO                   BEGIN
            K ← I DIV 75                                                   ;
            FORJ 74 DO D[J] ← (B[3,I+J] - V0)×SLOPE                        ;
            FORJ 74 DO D[J+75] ← (B[10,I+J] - V0)×SLOPE                    ;
            C[3,K] ← AVG(D, 150)                                    END    ;
```

```
FOR I ← 0 STEP 75 UNTIL M[4]-75 DO                              BEGIN
     K ← I DIV 75                                                     ;
     FORJ 74 DO D[J] ← (B[4,I+J] - V0)×SLOPE                          ;
     C[4,K] ← AVG(D, 75)                                        END   ;
FOR L ← 6,7,8,9 DO
FOR I ← 0 STEP 75 UNTIL MIN(M[L], M[L+14]) DO                   BEGIN
     K ← I DIV 75                                                     ;
     FORJ 74 DO D[J] ← (B[L,I+J] - V0)×SLOPE                          ;
     FORJ 74 DO D[J+75] ← (B[L+14,I+J] - V0)×SLOPE                    ;
     C[L,K] ← AVG(D, 150)                                       END   ;
FOR L ← 17, 18, 19, 28 DO
     FOR I ← 0 STEP 150 UNTIL M[L]     DO                       BEGIN
          K ← I DIV 150                                               ;
          FOR J ← 0 STEP 1 UNTIL 149 DO
               D[J] ← (B[L,I+J] - V0)×SLOPE                           ;
          C[L,K] ← AVG(D, 150)                                  END   ;
FOR L ← 10 STEP 1 UNTIL 16 DO                                   BEGIN
     FOR I ← 0 STEP 1 UNTIL M[L] DO
          B[L,I] ← (B[L,I] - V0)×SLOPE                                ;
     C[L,1] ← AVG(B[L,*], M[L])                                 END   ;
FOR I ← 0 STEP 1 UNTIL 29 DO
     T[I] ← B[0,I×25]                                                 ;
FOR L ← 11,13,14,15,16 DO
     P[L] ← RES(11500×C[L,1]/(V1-C[L,1]) - 665)                       ;
P[10] ← RES(4957.7×C[10,1]/(23.01 - C[10,1]) - 185.5)                 ;
P[12] ← RES(8431.6×C[12,1]/(V2 - C[12,1]) - 484.9)                    ;
WRITE (FIL3[NO], FMT3, LIST3)                                         ;
FOR I ← 0 STEP 1 UNTIL 4 DO                                     BEGIN
     P[17] ← 2.385×C[17,I]                                            ;
     P[18] ← 8.031×C[18,I]                                            ;
     P[19] ← 6.969×C[19,I]                                            ;
     P[28] ← C[28,I]                                                  ;
     WRITE (FIL4[NO], FMT7, LIST4)                                    ;
     FOR J ← 2×I, 2×I+1 DO                                  BEGIN
          P[5] ← C[3,J]                                               ;
          FOR L ← 6,7,8,9 DO
               P[L] ← C[L,J]                                          ;
```

```
P[4] ← IF C[4,J] > 3.58 THEN 681.11 - C[4,J]×89.46 ELSE
       IF C[4,J] > 1.0 THEN 213.89 - C[4,J]×47.81 ELSE 225.0;
T1 ← 200×ABS(P[6] - P[8])/(P[6] + P[8])                            ;
T2 ← 200×ABS(P[7] - P[9])/(P[7] + P[9])                            ;
WRITE (FIL3[NO], FMT4, LIST2)                                      ;
T1 ← T[K] DIV 3600                                                 ;
T2 ← (T[K] - T1×3600) DIV 60                                       ;
T3 ← T[K] - 3600×T1 - 60×T2                                        ;
WRITE (FIL4, FMT6, LIST5)                                          ;
CNT2 ← CNT2 +1                                                     ;
IF CNT2 > 50                                        THEN  BEGIN
     CNT2 ← 0                                                      ;
     PAGE2 ← PAGE2 +1                                              ;
     WRITE (FIL4[PAGE])                                            ;
     WRITE (FIL4, FMT2, PAGE2)                                     ;
     WRITE (FIL4, FMT1A)                                  END      ;
FOR K ← 3×J, 3×J+1, 3×J+2 DO                              BEGIN
     P[2] ← (-90.0 + (T[K] - 25200)/240)/57.29578                  ;
     P[2] ← COS(P[2]) - 0.070                                      ;
     IF P[2] ≠ 1.0                                  THEN
          P[3] ← 1.570796 - ARCTAN(P[2]/SQRT(1-P[2]*2))
                                                    ELSE
          P[3] ← 1.570796                                          ;
     P[3] ← P[3]×57.29578                                          ;
     C[5,K] ← C[5,K]×28.5/P[18]                                    ;
     IF C[5,K] < 1.32                               THEN
          C[5,K] ← 1.32                                            ;
     P[2] ← 62.44×(C[5,K] - 4.873 + 0.011×P[3])                    ;
     IF P[2] < -20.0                                THEN
          P[2] ← P[2] + 265.8                                      ;
     P[1] ← P[2] - P[3]                                            ;
     T1 ← T[K] DIV 3600                                            ;
     T2 ← (T[K] - T1×3600) DIV 60                                  ;
     T3 ← T[K] - 3600×T1 - 60×T2                                   ;
     WRITE (FIL3, FMT5, LIST1)                                     ;
     CNT1 ← CNT1 +1                                                ;
     IF CNT1 > 51                                   THEN  BEGIN
```

```
                 CNT1 ← 0                                             ;
                 PAGE1 ← PAGE1 + 1                                    ;
                 WRITE (FIL3[PAGE])                                   ;
                 WRITE (FIL3, FMT2, PAGE1)                            ;
                 WRITE (FIL3, FMT1)                      END          ;
                                                         END          ;
                                                         END    END   ;
END ;
L4:   WRITE (FIL3[PAGE])                                              ;
      WRITE (FIL3, FMT8, BLK)                                         ;
END.
```

```
COMMENT     DATA  REDUCTION PROGRAM FOR OBTAINING INTENSITY DATA AND
            POLARIZATION DATA  AT A GIVEN TIME.  THE PROGRAM WILL OBTAIN
            ABSOLUTE INTENSITIES IF THE REFERENCE GENERATOR WAS
            OPERATING PROPERLY, BUT IT WILL ALSO FURNISH RELATIVE
            INTENSITIES IF THE GENERATOR AND QUARTER WAVE PLATE DRIVE HAS
            STOPPED. INPUT DATA MUST BE IN THE FOLLOWING FORMAT:
            CARD 1     COLUMN 1-5          STARTING TIME
                       COLUMN 11-15        STOPPING TIME
                       COLUMN 21-24        WAVLENGTH INTERVAL FOR SAMPLES
            CARD 2     COLUMN 1-5          MIN. VALUE FOR "SHUTTER"
                       COLUMN 11-15        MAX. VALUE FOR "SHUTTER"
                       COLUMN 21-26        541F SHUTTER ATTENUATION (REAL)
                       COLUMN 31-36        541A SHUTTER ATTENUATION (REAL)
                       COLUMN 41-46        APPROX. GENERATOR FREQUENCY
            CARD 3     COLUMN 1-7          541F GEN. PHASE ANGLE (REAL)
                       COLUMN 11-17        541A GEN. PHASE ANGLE (REAL)
                       COLUMN 21-24        MINIMUM WAVELENGTH FOR 541A TUBE
                       COLUMN 26-29        MAXIMUM WAVELENGTH FOR 541F TUBE
                       COLUMN 31-33        COMPUTE OPTION FOR GEN. DATA
                       COLUMN 34-36        COMPUTE OPTION FOR PM DATA
                       COLUMN 37-41        PRINT OPTION FOR GEN. DATA
                       COLUMN 42-46        PRINT OPTION FOR PM DATA          ;
BEGIN INTEGER INTV, STRTTM, STPTM, N                                         ;
     FILE OUT FIL2 6 (5, 15),
              FPUNCH 0 (5, 15)                                               ;
     DEFINE HOTIME = (A[1].[30:17] + A[2].[30:17])/2 #                       ;
FORMAT OUT FMT1 ("MALFUNCTION OF AC GENERATOR"/"ONLY RELATIVE",
                " INTENSITIES ARE OBTAINED"),
            FMT2 (X1, I4),
            FMT3 (X8, E10.3, X13, 2(F4.2, X8), F5.1),
            FMT4 (X19, E10.3, 2(X8, F4.2), X9, F5.1),
            FMT5 (X72, 2(R9.2, X13)),
            FMT6 (X70, 2(X13, R9.2)),
            FMT7 ("PROCESSOR TIME ", F6.1, " SECONDS"/"IN-OUT TIME     ",
                F6.1, " SECONDS"),
            FMT8 (X47, "NATURAL SKY UV BACKGROUNDS"/X53,
                "INTENSITY DATA"/X48, "ALTITUDE: ", I3, ",", I1,
```

```
        "00 FEET"/X41, "STARTING SCATTER ANGLE: ", F5.1,
        " DEGREES"/X41, "STOPPING SCATTER ANGLE: ", F5.1,
        " DEGREES", X17,
        "AMB TEMP (F) ", I3/X93, "AVG. SOL. MON. ", F3.1/
        X87, "AVG. AZ. POS. (DEG.) ", I3//X14, "INTENSITY", X8,
        "DEGREE  OF", X16, "ANGLE OF", X11, "RELATIVE",
        X14, "STANDARD"/X13, "UW/SQ.CM./NM", X5,
        "POLARIZATION ELLIPTICITY POLARIZATION", X9,
        "INTENSITY", X13, "DEVIATION"/"LAMBDA", X5, "541F",
        X7, "541A   ", 2(X2, "541F  541A"),
        "   541F    541A", 2(X7, "541F", X7, "541A")///),
FMT9 ("STARTING TIME ",I2,":",I2,":"I2/"STOPPING TIME ",I2,
        ":",I2,":",I2//"MINIMUM WAVELENGTH ", I5/"MAXIMUM WA",
        "VELENGTH ", I5/I4," INTERVALS OF ", I3," ANGSTROMS"/),
FMT10(" PARITY ERROR -- STEP 1"),
FMT11(" PARITY ERROR -- STEP 2"),
FMT12(" PARITY ERROR -- STEP 3"),
FMT14(" PARITY ERROR -- STEP 5"),
FMT15("NO MINIMUM AND/OR MAXIMUM FOUND"/"MIN = 2000"/"MAX =",
        " 4000"),
FMT16("END OF FILE -- START TIME NOT FOUND"),
FMT17("C[", I2, "] = ", R10.3, X5, F10.6),
FMT18("FINAL APPROXIMATION FOR FREQUENCY OF AC",
        " GENERATOR WAS ", R12.5),
FMT19("PHASE ANGLE: ", F8.4/7(X25, "A[", I1, "] ", F10.5/)/),
FMT20("PHASE ANGLE: ", F8.4/"AMPLITUDE:  ", F8.4/
        3(X25, "A[", I1, "] ", F10.5/)/),
FMT21(10(F8.5, X3)),
FMT22("TUBE: ", I2/5(X10, "A[", I1, "] ", E12.5/)/),
FMT23("RUN DATE ", I2, "/", I2, "/", I2),
FMT24("RUN TIME ", I2, ":", I2, ":", I2),
FMT25("STANDARD DEVIATION ", R12.5/X3, 2(R12.5, X6), X2, I3),
FMT26(3(3(F10.5, X3))),
FMT27 (I4, X1, 2(F8.4, X2, F4.2, X2, 2(F5.3, X2)))            ;
FORMAT IN  FMTI1 (2(I5, X5),I4/2(I5, X5), 2(F6.2, X4), F6.3),
FMTI2(2(F7.4, X3), 2(I4, X1), 2I3, 2L5),
FMTI3 (2(F7.4, X3), F7.5)                                     ;
```

```
     REAL PI, PI2                                                    ;
REAL PROCEDURE ARCCOS (X)                                            ;
     VALUE X                                                         ;
     REAL X                                                          ;
     IF ABS(X) = 1.0                                        THEN
          ARCCOS ← 1.57079633×SIGN(X)                       ELSE
          ARCCOS ← 1.57079633 - ARCTAN(X/SQRT(1.0-X*2))              ;
COMMENT ARC TANGENT PROCEDURE WHOSE OUTPUT IS BETWEEN 0 AND 2×PI     ;
REAL PROCEDURE ATAN (A, B)                                           ;
     VALUE A, B                                                      ;
     REAL A, B                                                       ;
BEGIN REAL T                                                         ;
     IF B = 0                                               THEN
          T ← 1.57079633×SIGN(A)                            ELSE
          IF B < 0                                          THEN
               T ← 3.14159266 + ARCTAN(A/B)                 ELSE
               T ← ARCTAN (A/B)                                      ;
     IF T < 0                                               THEN
          ATAN ← T + 6.28318554                             ELSE
          ATAN ← T                                                   ;
END ATAN                                                             ;
     PROCEDURE DATE (FILEID)                                         ;
     FILE FILEID                                                     ;
BEGIN ALPHA D                                                        ;
     INTEGER I,J,K                                                   ;
     LABEL L1                                                        ;
     INTEGER ARRAY MO[1:12]                                          ;
     D ← TIME(0)                                                     ;
     FILL MO[*] WITH 31,28,31,30,31,30,31,31,30,31,30,31             ;
     K ← 10×D.[18:6] + D.[24:6]                                      ;
     J ← 100×D.[30:6] + 10×D.[36:6] + D.[42:6]                       ;
     IF J > (IF K MOD 4 = 0 THEN 366 ELSE 365)              THEN
          GO TO L1                                                   ;
     I ← 0                                                           ;
     FOR I ← I + 1 WHILE J > 0 DO                               BEGIN
          J ← J - MO[I]                                              ;
          IF  K MOD 4 = 0 AND I = 2                         THEN
```

```
                  J ← J - 1                                           END   ;
        I ← I - 1                                                         ;
        J ← J + MO[I]                                                     ;
        WRITE (FILEID, FMT23, I,J,K)                                      ;
L1:     K ← TIME(1) DIV 60                                                ;
        I ← K DIV 3600                                                    ;
        J ← (K - I×3600) DIV 60                                           ;
        K ← K - I×3600 - J×60                                             ;
        WRITE (FILEID, FMT24, I,J,K)                                      ;
END                                                                       ;
COMMENT CURVEFITTING PROCEDURE -- EXPLANATION OF OPTIONS
             OPTION #1      STRAIGHT CURVEFITTING
                    #2      STRAIGHT CURVEFITTING WITH STD. DEV. PRINTED
                    #3      SAME AS #2 BUT WITH INPUT DATA LISTED
                    #4      CURVEFITTING AND REJECTION OF DATA POINTS
                            LYING OUTSIDE T STD. DEVIATIONS. THE STD. DEV.
                            IS CALCULATED AND PRINTED
                    #5      SAME AS 4 BUT WITH INPUT DATA LISTED
                    #6      SAME AS #3 BUT WITH NO LISTING
                    #7      SAME AS #5 BUT WITH NO LISTING                ;
PROCEDURE COMPUTE (N,M,X,T,A,E,O,FILEID)                                  ;
        VALUE N,M,T,O                                                     ;
       FILE FILEID                                                        ;
        INTEGER N,M,O                                                     ;
        ARRAY X[0,0], A[0]                                                ;
        INTEGER ARRAY E[0]                                                ;
        REAL T                                                            ;
BEGIN   INTEGER I,J,K,L,P                                                 ;
        REAL T1, T2, S, D                                                 ;
        BOOLEAN ZZ                                                        ;
        LABEL L1, L2                                                      ;
        ARRAY Z[0:M-1,0:M], B,C[0:N-1]                                    ;
        LIST LIST1 (S, T1, T2, P),
            LIST2 (FOR I←0 STEP 1 UNTIL P-1 DO [X[M,I],B[I],C[I]])        ;
        ZZ ← FALSE                                                        ;
        E[0] ← P ← N                                                      ;
        FOR I ← 1 STEP 1 UNTIL N DO
```

```
          E[I] ← I-1                                                              ;
L1:    T1 ← T2 ← 0                                                                ;
       FOR I ← 0 STEP 1 UNTIL M-1 DO
       FOR J ← 0 STEP 1 UNTIL M DO                                        BEGIN
          Z[I,J] ← 0                                                              ;
          FOR K ← 0 STEP 1 UNTIL P-1 DO
             Z[I,J] ← Z[I,J] + X[I,K]×X[J,K]                              END    ;
       FOR K ← M STEP -1 UNTIL 0 DO                                       BEGIN
          D ← 0                                                                   ;
          FOR I ← 1 STEP 1 UNTIL K DO
             IF ABS(Z[I-1,0]) > D                          THEN    BEGIN
                L ← I-1                                                           ;
                D ← ABS(Z[L,0])                                    END            ;
          IF L > 0                                         THEN
             FOR J ←0 STEP 1 UNTIL K DO                            BEGIN
                D ← Z[L,J]                                                        ;
                Z[L,J] ← Z[0,J]                                                   ;
                Z[0,J] ← D                                         END            ;
          FOR I ← 0 STEP 1 UNTIL M-1 DO
             A[I] ← Z[I,0]                                                        ;
          FOR J ← 1 STEP 1 UNTIL K DO                              BEGIN
             D ← Z[0,J]/A[0]                                                      ;
             FOR I ← 1 STEP 1 UNTIL M-1 DO
                Z[I-1,J-1] ← Z[I,J] - A[I]×D                                      ;
             Z[M-1,J-1] ← D                                        END    END    ;
       IF O = 1                                            THEN
          GO TO L2                                                                ;
       FOR I ← 0 STEP 1 UNTIL P-1 DO                                      BEGIN
          B[I] ← A[0]×X[0,I]                                                      ;
          FOR J ← 1 STEP 1 UNTIL M-1 DO
             B[I] ← B[I] + A[J]×X[J,I]                                            ;
          C[I] ←X[M,I] - B[I]                                                     ;
          T1 ← T1 + C[I]                                                          ;
          T2 ← T2 + C[I]*2                                                END    ;
       T1 ← T1/P                                                                  ;
       T2 ← T2/P                                                                  ;
       S ← SQRT (T2 - T1*2)                                                       ;
```

```
      IF O > 3 AND O < 6                                         THEN
           WRITE (FILEID, FMT26, LIST2)                                            ;
      IF O < 6                                                   THEN
           WRITE (FILEID, FMT25, LIST1)                                            ;
      IF O = 3 OR O = 5 OR O = 7                                 THEN        BEGIN
           IF ZZ                                                 THEN
                GO TO L2                                                           ;
           ZZ ← TRUE                                                               ;
           P ← 0                                                                   ;
           FOR I ← 0 STEP 1 UNTIL N-1 DO
                IF ABS(C[I]) < T×S                               THEN  BEGIN
                     FOR J ← 0 STEP 1 UNTIL M DO
                          X[J,P] ← X[J,I]                                          ;
                     E[P+1] ← I                                                    ;
                     P ← P + 1                                         END         ;
           E[0] ← P                                                                ;
      GO TO L1                                                             END     ;
L2:END COMPUTE                                                                     ;
      DATE (FIL2)                                                                  ;
      PI ← 3.1415926536                                                            ;
      PI2 ← 1.5707963268                                                           ;
BEGIN INTEGER I, SCAN, TEMP, AZPOS                                                 ;
      FILE FDISK1 DISK SERIAL [1:1010] (1, 5, 30),
           FDISK2 DISK SERIAL [1:600] (1, 2, 30),
           FDISK3 DISK RANDOM [1:6000] (1, 3, 30)                                  ;
      FILE IN   FIL1 (2, 10),
                FMDATA (1, 1023),
                ALTTEMP "CLPNTS" "ALTTEMP" (1, 1010, 8),
                CONSTS "CLPNTS" "CONSTS" (1, 1010, 8),
                PMAMP "CLPNTS" "PMAMP" (1, 1001)                                   ;
      FILE OUT  FTAPE 2 (3, 1023, INTV×6, SAVE 5),
                FTP1 2 (2, 920, 9, SAVE 2)                                         ;
      ARRAY SHUT,ZETA,ETA[0:1], B,C1[0:8], TM[0:1,0:3+(STPTM-STRTTM)
          DIV 20]                                                                  ;
      INTEGER ARRAY OPT[0:2]                                                       ;
      BOOLEAN ARRAY TEST[0:1]                                                      ;
      REAL LAMBDA0, LAMBDA1, SHUT0, SHUT1, OMGA,
```

```
          LMB1, LMB2, ANG1, ANG2, STER, SOLMON                                 ;
      LABEL L1, L2, L3                                                         ;
      SWITCH FILE SWFIL ← CONSTS, ALTTEMP                                      ;
LIST  LIST1 (STRTTM, STPTM, INTV, SHUT0, SHUT1, SHUT[0], SHUT[1], OMGA),
      LIST2 (ZETA[0],ZETA[1], LMB1,LMB2, OPT[0],OPT[1],
             TEST[0], TEST[1]),
      LIST3 (ETA[0], ETA[1], STER)                                             ;
      PROCEDURE REED (F, N, A, LE, LP)                                         ;
           FILE F                                                              ;
           INTEGER N                                                           ;
           ARRAY A[0]                                                          ;
           LABEL LE, LP                                                        ;
BEGIN READ (F[NO], 1, A[*])[LE:LP]                                             ;
      N ← A[0].[6:12]                                                          ;
      READ (F, N, A[*])
END                                                                            ;
      READ (FIL1, FMTI1, LIST1)                                                ;
      READ (FIL1, FMTI2, LIST2)                                                ;
      READ (FIL1, FMTI3, LIST3)                                                ;
      FOR I ← 0, 1 DO                                                      BEGIN
           READ (SWFIL[I], 8, B[*])                                            ;
COMMENT    INITIALIZATION BLOCK WHERE ALL CONSTANTS TO BE USED BY THE
           PROGRAM ARE PLACED ON THE DISK FROM TAPE, FROM WHICH THEY
           WILL EVENTUALLY BE READ                                             ;
BEGIN INTEGER J, K                                                             ;
      ARRAY C[0:4]                                                             ;
      IF I = 0                                               THEN          BEGIN
           C1[0] ← C1[2] ← B[6]                                                ;
           C1[1] ← C1[3] ← B[7]                                                ;
           C1[4] ← B[2]/2                                                      ;
           FOR J ← 0 STEP 1 UNTIL B[2]-1 DO                                BEGIN
                READ (CONSTS, 5, C[*])                                         ;
                WRITE (FDISK1, 5, C[*])                                    END ;
           REWIND (CONSTS)                                                     ;
           REWIND (FDISK1)                                                 END
                                                             ELSE          BEGIN
           C1[5] ← B[4]                                                        ;
```

```
        C1[6] ← B[5]                                                      ;
        FOR J ← 0 STEP 1 UNTIL B[2]-1 DO                            BEGIN
             READ (ALTTEMP, 2, C[*])                                      ;
             WRITE (FDISK2, 2, C[*])                                END   ;
        REWIND (ALTTEMP)                                                  ;
        REWIND (FDISK2)                                             END   ;
END *                                                                     ;
                                                                    END   ;
     C1[7] ← 42120                                                        ;
     C1[8] ← 5                                                            ;
     CLOSE (FIL1, RELEASE)                                                ;
BEGIN INTEGER I,J,K,L,M,N,P,R                                             ;
     REAL A1,B1,S,OMEGA,OMIN,OMAX                                         ;
     ARRAY G[0:250], GEN,LMB[0:6×INTV], PM[0:1,0:6×INTV], C[0:21],
           PMCAL[0:1,0:1001]                                              ;
     ALPHA ARRAY A[0:1022]                                                ;
     LIST LIST1 (FOR A1 ← STRTTM, STPTM DO [I ← A1 DIV 3600, J ← (A1 -
                 3600×I) DIV 60, A1 - 3600×I - 60×J], LAMBDA0,LAMBDA1,
                 M, INTV)                                                 ;
     BOOLEAN Z                                                            ;
     LABEL L1,L2,L3,L4,L6,L7,L8,L9,L10,LE1,LE2,LP1,LP2,LP3,LP5            ;
     READ (PMAMP, 8, A[*])                                                ;
     FOR I ← 0,1 DO
          READ (PMAMP, 1001, PMCAL[I,*])                                  ;
     L ← 0                                                                ;
     Z ← FALSE                                                            ;
     LAMBDA0 ← 1                                                          ;
     LAMBDA1 ← 0                                                          ;
COMMENT    DETERMINATION OF MINIMUM AND MAXIMUM WAVE LENGTH               ;
     DO READ (FMDATA, 4, A[*])[LE1:LP1] UNTIL A[3].[1:13] = 0             ;
     DO READ (FMDATA, 4, A[*])[LE1:LP1] UNTIL A[3].[1:13] ≠ 0             ;
     SPACE (FMDATA, -2)                                                   ;
     REED (FMDATA, N, A, LE1, LP1)                                        ;
     FOR I ← 7 STEP 4 UNTIL N-1 DO
          IF A[I-4].[1:13] = 0 AND A[I].[1:13] ≠ 0   THEN           BEGIN
               LAMBDA0 ← A[I].[1:13]                                      ;
               GO TO L1                                             END   ;
```

```
L1:    DO READ (FMDATA, 4, A[*])[LE2:LP2] UNTIL A[3].[1:13] = 0            ;
       SPACE (FMDATA, -2)                                                  ;
       REED (FMDATA, N, A, LE1, LP1)                                       ;
       FOR I ← 7 STEP 4 UNTIL N-1 DO
            IF A[I-4].[1:13] ≠ 0 AND A[I].[1:13] = 0   THEN          BEGIN
                 LAMBDA1 ← A[I-4].[1:13]                                   ;
                 GO TO L2                                            END   ;
L2:    P ← (N-3) DIV 4 - 1                                                 ;
       FOR I ← 0 STEP 1 UNTIL P DO
            G[I] ← A[4×I+3].[14:11]                                        ;
       REWIND (FMDATA)                                                     ;
       M ← (LAMBDA1 - LAMBDA0)/INTV                                        ;
       WRITE (FIL2, FMT9, LIST1)                                           ;
COMMENT     POSITIONING OF TAPE TO STARTING TIME                           ;
L3:    READ (FMDATA, 3, A[*])[LE2:LP2]                                     ;
       IF A[1].[30:17]≠A[2].[30:17] OR (A[1].[30:17]<STRTTM)
                                                       THEN
            GO TO L3                                                       ;
       SPACE (FMDATA, -1)                                                  ;
       SCAN ← 0                                                            ;
COMMENT     FILLING OF SCRATCH TAPE WITH DATA TO BE USED IN THE
            CALCULATIONS                                                   ;
L4:    REED (FMDATA, N, A,  L8, LP3)                                       ;
       K ← 3                                                               ;
       IF A[3].[1:13] ≠ 0                              THEN          BEGIN
            TM[0, SCAN] ← HOTIME                                           ;
            J ← (A[3].[1:13] - LAMBDA0) DIV INTV                     END
                                                       ELSE          BEGIN
            J ← 0                                                          ;
            IF A[N-1].[1:13] ≠ 0                       THEN  BEGIN
                 DO K←K+4 UNTIL A[K].[1:13] ≠ 0                            ;
                 TM[0, SCAN] ← HOTIME                        END           ;
            IF A[N-1].[1:13] = 0                       THEN
                 GO TO L4                                            END   ;
L6:    FOR I ← K STEP 4 UNTIL N-1 DO                                 BEGIN
            L ← L+1                                                        ;
            LMB[L] ← A[I+1].[1:13]                                         ;
```

```
          GEN[L] ← A[I].[14:11]                                             ;
          PM[0,L] ← PMCAL[0,A[I+2].[14:11]/2]                               ;
          PM[1,L] ← PMCAL[0,A[I+3].[14:11]/2]                               ;
          IF A[I+1].[14:11]>SHUT0 AND A[I+1].[14:11]<SHUT1
                                                     THEN
               FOR R ← 0,1 DO
                    PM[R,L] ← PM[R,L]×SHUT[R]                               ;
          IF A[I].[1:13]=0 OR (A[I].[1:13]-LAMBDA0) DIV INTV>J
                                                     THEN  BEGIN
               J ← J+1                                                      ;
               GEN[0] ← PM[0,0] ← PM[1,0] ← LMB[0] ← L+1                    ;
               WRITE (FTAPE, L+1, LMB[*])                                   ;
               WRITE (FTAPE, L+1, GEN[*])                                   ;
               WRITE (FTAPE, L+1, PM[0,*])                                  ;
               WRITE (FTAPE, L+1, PM[1,*])                                  ;
               L ← 0                                                        ;
COMMENT   DETERMINATION OF END OF SCAN CONDITIONS                           ;
               IF A[I].[1:13] = 0                    THEN  BEGIN
                    TM[1, SCAN] ← HOTIME                                    ;
                    SCAN ← SCAN + 1                                         ;
                    GO TO L4                               END              ;
               IF J > M                              THEN
                    J ← 0                                  END    END       ;
L7:   REED (FMDATA, N, A,  L8, LP5)                                         ;
      K ← 3                                                                 ;
          IF (A[1].[30:17]=A[2].[30:17]) AND (A[1].[30:17]<STPTM)
                                                     THEN
               GO TO L6                                                     ;
      TM[1, SCAN] ← HOTIME                                                  ;
L8:   REWIND (FTAPE)                                                        ;
      LOCK (FMDATA, RELEASE)                                                ;
COMMENT   CALCULATION OF APPROXIMATE FREQUENCY OF REF. GEN.                 ;
      OMIN ← 0.9×OMGA                                                       ;
      OMAX ← 1.1×OMGA                                                       ;
      S ← (OMAX - OMIN)/10.0                                                ;
L9:   K ← 0                                                                 ;
      FOR OMEGA ← OMIN STEP S UNTIL OMAX DO                          BEGIN
```

```
         A1 ← B1 ← 0                                                       ;
         FOR I ← 0 STEP 1 UNTIL P DO                             BEGIN
              A1 ← A1 + G[I]×COS(I×OMEGA)                                  ;
              B1 ← B1 + G[I]×SIN(I×OMEGA)                        END       ;
         C[K] ← A1*2 + B1*2                                                ;
         IF TEST[0]                                        THEN
              WRITE (FIL2, FMT17, K, C[K], OMEGA)                          ;
         K ← K+1                                                     END   ;
    A1 ← C[0]                                                              ;
    FOR I ← 1 STEP 1 UNTIL K-1 DO
         IF C[I] > A1                                      THEN      BEGIN
              A1 ← C[I]                                                    ;
              OMGA ← OMIN + I×S                                      END   ;
    IF (OMAX - OMIN)/OMGA < 0.01                           THEN
         GO TO L10                                                         ;
    OMIN ← OMGA - S                                                        ;
    OMAX ← OMGA + S                                                        ;
    S ← S/10.0                                                             ;
    GO TO L9                                                               ;
COMMENT  PARITY ACTIONS AND LABELS                                         ;
LP1:  WRITE (FIL2, FMT10)                                                  ;
      GO TO L1                                                             ;
LP2:  WRITE (FIL2, FMT11)                                                  ;
      GO TO L3                                                             ;
LP3:  WRITE (FIL2, FMT12)                                                  ;
      GO TO L4                                                             ;
LP5:  WRITE (FIL2, FMT14)                                                  ;
      GO TO L7                                                             ;
LE1:  WRITE (FIL2, FMT15)                                                  ;
      LAMBDA0 ← 2000                                                       ;
      LAMBDA1 ← 4000                                                       ;
      GO TO L2                                                             ;
LE2:  WRITE (FIL2, FMT16)                                                  ;
      GO TO L10                                                            ;
L10:  IF TEST[0]                                           THEN
          WRITE (FIL2, FMT18, OMGA)                                        ;
END *                                                                      ;
```

```
       SCAN ← 0                                                              ;
L1:    SPACE (FTAPE, 4)[L2]                                                  ;
       READ (FTAPE[NO], *, N)[L2]                                            ;
BEGIN  INTEGER I,J,K,Q                                                       ;
       REAL PHI, AMP, R, OMEGA, A1, B1, LAMBDA, BETA2, INT,POL,ELL,PHI2      ;
       INTEGER ARRAY B, C[0:N+INTV]                                          ;
       ARRAY GEN[0:N+INTV], PM[0:1,0:N+INTV], A[0:7], X[0:7,0:N+INTV],
             THETA2[0:N+INTV], LMB[0:N+INTV], TAPE[0:9]                      ;
       LABEL L1, L2, L3, L4                                                  ;
       LIST LIST1 (PHI, FOR I ← 0 STEP 1 UNTIL 6 DO [I, A[I]]),
            LIST2 (PHI, AMP, FOR I ← 0,1,2 DO [I, A[I]]),
            LIST3 (FOR I ← 0 STEP 1 UNTIL N-2 DO THETA2[I]),
            LIST4 (Q, FOR I ← 0 STEP 1 UNTIL 4 DO [I, A[I]])                 ;
       LAMBDA ← 999999                                                       ;
L1:    READ (FTAPE[NO], *, N)[L4]                                            ;
       READ (FTAPE, N, LMB[*])                                               ;
       READ (FTAPE, N, GEN[*])                                               ;
       FOR I ← 0,1 DO
            READ (FTAPE, N, PM[I,*])                                         ;
COMMENT     DETERMINATION OF THE ANGLE OF THE QUARTER WAVE PLATE             ;
       FOR I ← 1 STEP 1 UNTIL N-1 DO                                    BEGIN
            X[0,I-1] ← 1.0                                                   ;
            X[1,I-1] ← SIN(OMGA×(I))                                         ;
            X[2,I-1] ← COS(OMGA×(I))                                         ;
            FOR K ← 1,2 DO                                         BEGIN
                 X[2×K+1,I-1] ← X[1,I-1]×((I)*K)                         ;
                 X[2×K+2,I-1] ← X[2,I-1]×((I)*K)                   END       ;
            X[7,I-1] ← GEN[I]                                           END  ;
       COMPUTE (N-2, 7, X, 1.0, A, B, OPT[0], FIL2)                          ;
       PHI ← ATAN (A[2], A[1])                                               ;
       IF TEST[0]                                           THEN
            WRITE (FIL2, FMT19, LIST1)                                       ;
       IF ABS (A[4]-A[3]) > 0.001                           THEN
            IF ABS (A[2]) > ABS (A[1])                      THEN
                 OMGA ← OMGA - A[3]/A[2]                    ELSE
                 OMGA ← OMGA + A[4]/A[1]                                     ;
       FOR I ← 0 STEP 1 UNTIL B[0]-1 DO                                 BEGIN
```

```
          X[1,I] ← SIN(OMGA×B[I+2])                                   ;
          X[2,I] ← COS(OMGA×B[I+2])                                   ;
          X[3,I] ← GEN[B[I+2]]                               END      ;
     COMPUTE (B[0], 3, X, 1.0, A, C, OPT[0], FIL2)                    ;
     PHI ← ATAN (A[2], A[1])                                          ;
     AMP ← SQRT (A[1]*2 + A[2]*2)                                     ;
     IF TEST[0]                                  THEN
          WRITE (FIL2, FMT20, LIST2)                                  ;
     IF AMP < 250.0                              THEN        BEGIN
          WRITE (FIL2, FMT1)                                          ;
          GO TO L2                                           END      ;
     FOR I ← 0 STEP 1 UNTIL N-2 DO
          THETA2[I] ← PHI + I×OMGA                                    ;
     IF TEST[0]                                  THEN
          WRITE (FIL2, FMT21, LIST3)                                  ;
     A1 ← 0                                                           ;
     FOR I ← 1 STEP 1 UNTIL N-1 DO
          A1 ← A1 + LMB[I]                                            ;
     A1 ← A1/(N-1)                                                    ;
     IF LAMBDA > A1                              THEN        BEGIN
          WRITE (FIL2[PAGE])                                          ;
          SCAN ← SCAN + 1                                             ;
          FOR I ← 0,1 DO                                 BEGIN
               J ← (TM[I,SCAN] - C1[7])/C1[8]                         ;
          I ← ((TM[0,SCAN]+TM[1,SCAN])/2 - C1[5])/C1[6]               ;
          READ (FDISK2[I], 2, A[*])                                   ;
          I ← ENTIER (A[0])                                           ;
          J ← (A[0] - I)×10                                           ;
          IF J = 10                              THEN    BEGIN
               J ← 0                                                  ;
               I ← I + 1                                 END          ;
          WRITE (FIL2, FMT8, I,J,ANG1,ANG2,A[1],SOLMON,AZPOS)   END   ;
                                                         END          ;
     LAMBDA ← A1                                                      ;
     WRITE (FIL2[NO], FMT2, LAMBDA)                                   ;
     IF OPT[1] < 6                               THEN
          WRITE (FIL2[DBL])                                           ;
```

```
     FOR Q ← 0 STEP 1 UNTIL 1 DO                                    BEGIN
          IF LAMBDA > LMB2                                    THEN
               Q ← 1                                                       ;
COMMENT   CURVEFITTING THE PM CURRENT TO OBTAIN ITS FOURIER COMPONENTS;
          FOR I ←1 STEP 1 UNTIL N-1 DO                             BEGIN
               BETA2 ← 2×ZETA[Q] + THETA2[I-1]                            ;
               X[0,I-1] ← 1.0                                              ;
               X[1,I-1] ← SIN(BETA2)                                       ;
               X[2,I-1] ← COS(BETA2)                                       ;
               X[3,I-1] ← SIN(2×BETA2)                                     ;
               X[4,I-1] ← COS(2×BETA2)                                     ;
               X[5,I-1] ← PM[Q,I]                                  END    ;
          COMPUTE (N-1, 5, X, 0.9, A, B, OPT[1], FIL2)                    ;
          IF TEST[1]                                          THEN
               WRITE (FIL2, FMT22, LIST4)                                  ;
          IF Q = 0                                            THEN
               I  ← (LAMBDA - C1[0])/C1[1]                    ELSE
               I  ← (LAMBDA - C1[2])/C1[3] + C1[4]                         ;
          READ (FDISK1[I],  5, GEN[*])                                     ;
          INT ← A[0]×GEN[0] + A[4]×GEN[1]                                  ;
          POL ← GEN [2]×SQRT(A[4]*2 + A[3]*2)/INT                          ;
          PHI2 ← ATAN (A[3], A[4])                                         ;
          ELL ← A[2]×GEN[3]/INT + GEN[4]×POL×SIN(PHI2)                     ;
          IF INT < 0                                          THEN
               INT ← 0                                        ELSE
               INT ← INT/STER                                              ;
          PHI2 ← PHI2/2 + ETA[Q]                                           ;
          TAPE[0] ← LAMBDA                                                 ;
          TAPE[4×Q+1] ← IF INT > 0 THEN LN(INT) ELSE 0                     ;
          TAPE[4×Q+2] ← POL                                                ;
          TAPE[4×Q+3] ← ELL                                                ;
          TAPE[4×Q+4] ← PHI2                                               ;
          IF Q = 0                                            THEN
               WRITE (FIL2[NO], FMT3, INT,POL,ELL,PHI2)
                                                              ELSE
               WRITE (FIL2[NO], FMT4, INT,POL,ELL,PHI2)                    ;
          IF OPT[1] < 6                                       THEN
```

```
                 WRITE (FIL2[DBL])                                                  ;
            IF LAMBDA < LMB1                                  THEN
                 Q ← 1                                                      END     ;
      GO TO L3                                                                      ;
COMMENT -- AVERAGING PORTION FOR WHEN THE AC GENERATOR IS NOT
            FUNCTIONING                                                             ;
L2:   WRITE (FIL2[NO], FMT2, LAMBDA)                                                ;
      FOR Q ← 0 STEP 1 UNTIL 1 DO                                           BEGIN
            IF LAMBDA < LMB2                                  THEN
                 Q ← 1                                                              ;
            A1 ← B1 ← 0                                                             ;
            FOR I ← 1 STEP 1 UNTIL N-1 DO                           BEGIN
                 A1 ← A1 + PM[Q,I]                                                  ;
                 B1 ← B1 + PM[Q,I]*2                                END             ;
            B1 ← B1/(N-1)                                                           ;
            INT ← A1/(N-1)                                                          ;
            POL ← SQRT (B1 - INT*2)                                                 ;
            IF Q = 0                                          THEN
                 WRITE (FIL2[NO], FMT5, INT, POL)             ELSE
                 WRITE (FIL2[NO], FMT6, INT, POL)                                   ;
            IF LAMBDA < LMB1                                  THEN
                 Q ← 1                                                      END     ;
L3:   WRITE (FIL2)                                                                  ;
      WRITE (FTP1, 9, TAPE[*])                                                      ;
      GO TO L1                                                                      ;
L4:   END                                                                           ;
      GO TO L1                                                                      ;
L2:   REWIND (FTP1)                                                                 ;
      WHILE TRUE DO                                                         BEGIN
            READ (FTP1, 9, B[*])[L3]                                                ;
            WRITE (FPUNCH, FMT27, FOR I←0 STEP 1 UNTIL 8 DO B[I])           END     ;
L3:   CLOSE (FTP1, PURGE)                                                           ;
      WRITE (FIL2, FMT7, TIME(2)/60, TIME(3)/60)                                    ;
      CLOSE (FTAPE, PURGE)                                                          ;
END                                                                                 ;
END .
```

```
COMMENT -- CALIBRATION PROGRAM                                        ;
BEGIN INTEGER M, SMP, U, SENS                                         ;
      REAL RNTIM1, RNTIM2, RNTIM3, PI, PI2                            ;
      FILE FIL2 6 (2,15)                                              ;
      FILE FILIN1 (2,10)                                              ;
      FILE IN FLDT1 "MULF2" "FTP1" (1,550)                            ;
      FILE FIL4 2 (1,10, SAVE 10)                                     ;
REAL PROCEDURE ARCCOS(X)      ; VALUE X; REAL X                       ;
BEGIN REAL T  ; IF ABS(X)=1.0 THEN T←1.57079633×SIGN(X) ELSE
           T ← ARCTAN(X/SQRT(1 - X*2))                                ;
           ARCCOS ← 1.57079633 - T END ARCCOS                         ;
REAL PROCEDURE ATAN(A,B) ; VALUE A,B ; REAL A,B                       ;
BEGIN REAL T; IF B=0 THEN T←1.57079633×SIGN(A) ELSE IF B<0 THEN
          T ← 3.14159266 +ARCTAN(A/B) ELSE T←ARCTAN(A/B)              ;
      IF T<0 THEN T←T+6.28318554                                      ;
      ATAN ← T END                                                    ;
FORMAT IN : FMTI1(3I4),
          FMTI2(4I6/7F10.6),
          FMTI3(6I2,X8,9L5),
          FMTI4(10I6)                                                 ;
FORMAT OUT FMT1("DATA PROCESSOR TIME: ", F8.2, " SECONDS"),
          FMT2("INPUT - OUTPUT TIME: ", F8.2, " SECONDS",             ;
          FMT3("TUBE LAM        CI          RR         RP       DELTA   ",
               "   ZETA       ETA")                                   ;
STREAM PROCEDURE WORD(A,B)                                            ;
BEGIN SI ← A; DI← B; SI ← SI - 8; DS ← 8 CHR      END                 ;
PROCEDURE DATE(FILEID)                                                ;
      FILE FILEID                                                     ;
BEGIN ALPHA D; INTEGER I,DAY,YR ; LABEL L1; INTEGER ARRAY MO[1:12]    ;
      FORMAT OUT FMT("RUN DATE ", I2,"/",I2,"/",I2)                   ;
      FILL MO[*] WITH 31,28,31,30,31,30,31,31,30,31,30,31             ;
      D ←TIME(0)                                                      ;
      YR ← 10×D.[18:6] + D.[24:6]                                     ;
      DAY ← 100×D.[30:6] + 10×D.[36:6] + D.[42:6]                     ;
      IF DAY > (IF YR MOD 4 = 0 THEN 366 ELSE 365)    THEN
          GO TO L1                                                    ;
      I ←0                                                            ;
```

```
      FOR I ← I + 1 WHILE DAY > 0 DO                                            BEGIN
           DAY ←DAY - MO[I]                                                          ;
           IF YR MOD 4 = 0 THEN IF I=2 THEN DAY ← DAY-1                         END  ;
      I ← I-1                                                                         ;
      DAY ←DAY + MO[I]                                                                ;
      WRITE (FILEID, FMT, I,DAY,YR)                                                   ;
L1:END DATE                                                                           ;
PROCEDURE DAYTIM(FILEID)                                                              ;
      FILE FILEID                                                                     ;
BEGIN INTEGER I1, I2, I3, T                                                           ;
      FORMAT OUT FMT("RUN TIME ", I2,":",I2,":",I2)                                   ;
      T ← TIME(1) DIV 60                                                              ;
      I1 ← T DIV 3600                                                                 ;
      I2 ← (T - I1×3600) DIV 60                                                       ;
      I3 ← T - I1×3600 - I2×60                                                        ;
      WRITE (FILEID, FMT, I1, I2, I3)                                                 ;
END                                                                                   ;
PROCEDURE SORT(X,K)                                                                   ;
      VALUE K                                                                         ;
      INTEGER K                                                                       ;
      ARRAY X[0]                                                                      ;
BEGIN INTEGER I,J,L                                                                   ;
      ARRAY F[0:K-1]                                                                  ;
      FOR J ← K-1 STEP -1 UNTIL 0 DO                                            BEGIN
           F[J] ←X[0]                                                                 ;
           FOR I ← 1 STEP 1 UNTIL K-1 DO
                IF X[I] > F[J] THEN                                       BEGIN
                     F[J] ← X[I]                                                      ;
                     L ← I                                                END         ;
           X[L] ← 0                                                              END  ;
      FOR I ← 0 STEP 1 UNTIL K-1 DO
           X[I] ← F[I]                                                                ;
END                                                                                   ;
PROCEDURE COMPUTE(N,M,X,T,A,E,U,FILEID)                                               ;
      VALUE N,M,T,U                                                                   ;
      FILE FILEID                                                                     ;
      INTEGER N,M,U                                                                   ;
```

```
     ARRAY X[0,0], A[0]                                          ;
     INTEGER ARRAY E[0]                                          ;
     REAL T                                                      ;
BEGIN INTEGER I,J,P, K,L                                         ;
     REAL T1, T2, S, D                                           ;
     BOOLEAN ZZ                                                  ;
     LABEL L1, L2                                                ;
     ARRAY Z[0:M-1,0:M]                                          ;
     ARRAY B,C[0:N-1]                                            ;
     LIST LIST1(S, T1, T2, P),
     LIST2(FOR I ← 0 STEP 1 UNTIL P-1 DO [X[M,I],B[I],C[I]])     ;
     FORMAT OUT FMT1("STANDARD DEVIATION ",F10.5/X3,2(E12.5,X6),X2,I3),
                FMT2(3(3(F10.5,X3)))                             ;
     ZZ ← FALSE                                                  ;
     E[0] ←P ← N                                                 ;
     FOR I ← 1 STEP 1 UNTIL N DO
          E[I] ← I-1                                             ;
L1:  T1 ← T2 ← 0                                                 ;
     FOR I ← 0 STEP 1 UNTIL M-1 DO
     FOR J ← 0 STEP 1 UNTIL M DO                             BEGIN
          Z[I,J] ← 0                                             ;
          FOR K ← 0 STEP 1 UNTIL P-1 DO
               Z[I,J] ← Z[I,J] + X[I,K]×X[J,K]               END ;
     FOR K ← M STEP -1 UNTIL 0 DO                            BEGIN
          D ← 0                                                  ;
          FOR I ← 1 STEP 1 UNTIL K DO
               IF ABS(Z[I-1,0]) > D                 THEN  BEGIN
                    L ← I-1                                      ;
                    D ← ABS(Z[L,0])                       END    ;
          IF L>0                                    THEN
               FOR J ← 0 STEP 1 UNTIL K DO                BEGIN
                    D ← Z[L,J]                                   ;
                    Z[L,J] ← Z[0,J]                              ;
                    Z[0,J] ← D                            END    ;
          FOR I ← 0 STEP 1 UNTIL M-1 DO
               A[I] ← Z[I,0]                                     ;
          FOR J ← 1 STEP 1 UNTIL K DO                     BEGIN
```

```
              D ← Z[0,J]/A[0]                                          ;
              FOR I ← 1 STEP 1 UNTIL M-1 DO
                   Z[I-1,J-1] ← Z[I,J] - A[I]×D                        ;
              Z[M-1,J-1] ← D                              END    END   ;
     IF U = 1 THEN GO TO L2                                            ;
     FOR I ←0 STEP 1 UNTIL P-1 DO                                BEGIN
          B[I] ← A[0]×X[0,I]                                           ;
          FOR J ← 1 STEP 1 UNTIL M-1 DO
              B[I] ← B[I] + A[J]×X[J,I]                                ;
          C[I] ← X[M,I] - B[I]                                         ;
          T1 ← T1 + C[I]                                               ;
          T2 ← T2 + C[I]*2                                       END   ;
     T1 ← T1/P                                                         ;
     T2 ← T2/P                                                         ;
     S ← SQRT(T2 - T1*2)                                               ;
     IF U >3 THEN WRITE (FILEID, FMT2, LIST2)                          ;
     WRITE (FILEID, FMT1, LIST1)                                       ;
     IF U = 3 OR U = 5 THEN                                      BEGIN
     IF ZZ THEN GO TO L2                                               ;
     ZZ ← TRUE                                                         ;
     P ← 0                                                             ;
     FOR I ← 0 STEP 1 UNTIL N-1 DO
          IF ABS(C[I]) < T×S                       THEN  BEGIN
              FOR J ← 0 STEP 1 UNTIL M DO
                   X[J,P] ← X[J,I]                                     ;
              E[P+1] ←I                                                ;
              P ← P + 1                                   END          ;
     E[0] ← P                                                          ;
     GO TO L1                                                    END   ;
L2:END COMPUTE                                                         ;
REAL  PROCEDURE CALCPM(T,X)                                            ;
      VALUE T,X ; REAL X; INTEGER T                                    ;
BEGIN INTEGER J ; REAL S1                                              ;
      ARRAY I[0:72], V[0:1,0:72]                                       ;
      FILL I[*] WITH 0,
      1.0@-4, 6.0@-4, 1.0@-3, 2.0@-3, 3.0@-3, 4.0@-3, 5.0@-3, 6.0@-3,
      7.0@-3, 8.0@-3, 9.0@-3, 1.0@-2, 1.1@-2, 1.2@-2, 1.3@-2, 1.4@-2,
```

```
1.5@-2, 1.6@-2, 1.7@-2, 1.8@-2, 1.9@-2, 2.0@-2, 2.2@-2, 2.4@-2,
2.6@-2, 2.8@-2, 3.0@-2, 3.5@-2, 4.0@-2, 4.5@-2, 5.0@-2, 5.5@-2,
6.0@-2, 7.5@-2, 8.0@-2, 8.5@-2, 9.0@-2, 9.5@-2, 1.0@-1, 1.5@-1,
2.0@-1, 2.5@-1, 3.0@-1, 3.5@-1, 4.0@-1, 4.5@-1, 5.0@-1, 5.5@-1,
6.0@-1, 6.5@-1, 7.0@-1, 7.5@-1, 8.0@-1, 8.5@-1, 9.0@-1, 9.5@-1,
1.0@-0, 1.1@-0, 1.2@-0, 1.3@-0, 1.4@-0, 1.5@-0, 1.6@-0, 1.7@-0,
1.8@-0, 1.9@-0, 2.0@-0, 2.1@-0, 2.2@-0, 2.3@-0, 2.4@-0, 2.5@-0      ;
FILL V[0,*] WITH 0.018,
0.040,  0.169,  0.270,  0.501,  0.769,  1.003,  1.240,  1.520,
1.710,  1.880,  2.010,  2.090,  2.130,  2.190,  2.225,  2.260,
2.296,  2.330,  2.351,  2.380,  2.400,  2.430,  2.460,  2.498,
2.520,  2.540,  2.560,  2.595,  2.630,  2.650,  2.690,  2.718,
2.740,  2.795,  2.802,  2.820,  2.850,  2.865,  2.880,  3.030,
3.100,  3.235,  3.280,  3.400,  3.465,  3.530,  3.570,  3.645,
3.690,  3.751,  3.810,  3.870,  3.930,  3.985,  4.040,  4.095,
4.150,  4.245,  4.350,  4.450,  4.550,  4.660,  4.750,  4.852,
4.951,  5.051,  5.179,  5.250,  5.340,  5.430,  5.470,  5.510       ;
FILL V[1,*] WITH 0.019,
0.038,  0.040,  0.245,  0.488,  0.750,  0.990,  1.210,  1.430,
1.630,  1.750,  1.860,  1.940,  1.995,  2.040,  2.070,  2.100,
2.128,  2.155,  2.180,  2.208,  2.230,  2.250,  2.296,  2.330,
2.360,  2.385,  2.400,  2.448,  2.480,  2.503,  2.540,  2.570,
2.595,  2.675,  2.690,  2.715,  2.738,  2.750,  2.765,  2.885,
2.980,  2.080,  3.165,  3.235,  3.288,  3.345,  3.370,  3.400,
3.500,  3.555,  3.610,  3.670,  3.720,  3.777,  3.845,  3.895,
3.920,  4.032,  4.128,  4.229,  4.318,  4.420,  4.500,  4.600,
4.695,  4.785,  4.880,  4.970,  5.058,  5.155,  5.249,  5.350       ;
IF T = 0 THEN S1←-0.7796+0.003196×X ELSE S1←-0.9028+0.003254×X      ;
IF S1<V[T,0] THEN S1←V[T,0] ELSE IF S1>V[T,72] THEN S1←V[T,72]      ;
J ← 0                                                                ;
DO J←J+9 UNTIL S1 ≤ V[T,J]                                           ;
DO J←J-1 UNTIL S1 ≥ V[T,J]                                           ;
CALCPM←I[J]+(I[J+1]-I[J])×(S1-V[T,J])/(V[T,J+1]-V[T,J])      END     ;
PI ← 3.14159265                                                      ;
PI2 ← PI/2                                                           ;
SENS ← 0                                                             ;
DATE (FIL2)                                                          ;
```

```
      DAYTIM(FIL2)                                                           ;
      WRITE (FIL2[DBL])                                                      ;
      READ (FILIN1, FMTI1, M, SMP, U)                                        ;
BEGIN INTEGER Q,M1,L,STRTTM,LEVEL,LAMBDA1, LAMBDA2                          ;
      REAL BKTIM, BRKTIM, OMGA, PSID                                         ;
      ARRAY PSI[0:M-1], ETAAR,ZETAAR[0:1]                                    ;
      INTEGER ARRAY N[0:M-1], OPT[0:5], LAMBDA[0:U-1]                        ;
      FILE FTAPE1 2 (1,3×SMP+4,SMP, SAVE 10)                                 ;
      BOOLEAN ARRAY Z[0:10]                                                  ;
      ALPHA TUBE                                                             ;
      LABEL L1                                                               ;
      LIST LISTI1(STRTTM,LEVEL,LAMBDA1,LAMBDA2,BRKTIM,OMGA,PSID,
           ETAAR[0],ETAAR[1], ZETAAR[0], ZETAAR[1]),
           LISTI2(FOR L←0 STEP 1 UNTIL 5 DO OPT[L],FOR L←0 STEP 1
                 UNTIL 8 DO Z[L]),
           LISTI3(FOR L ← 0 STEP 1 UNTIL U-1 DO LAMBDA[L])                   ;
      READ (FILIN1, FMTI2, LISTI1)                                           ;
      READ (FILIN1, FMTI3, LISTI2)                                           ;
      READ (FILIN1, FMTI4, LISTI3)                                           ;
      CLOSE (FILIN1, RELEASE)                                                ;
      FOR L ← 0 STEP 1 UNTIL M-1 DO                                    BEGIN
           N[L] ←SMP                                                         ;
           PSI[L] ← L×PSID×0.0174532925                                END   ;
      SORT (LAMBDA,U)                                                        ;
      WORD(FLDT1(0),M1)                                                      ;
BEGIN INTEGER I,J,K,P,R                                                      ;
      REAL S, OMEGA, OMIN, OMAX, A1, B1                                      ;
      ALPHA ARRAY A[0:M1+5]                                                  ;
      BOOLEAN Z1                                                             ;
      ARRAY G[0:M1/4], GEN[0:SMP-1], H[0:1,0:SMP-1], TIM[0:M-1],C[0:21]      ;
      INTEGER ARRAY BLOCK[0:M-1]                                             ;
      LABEL L1, L2, L3, L4, L5, L6, L7, LE1, LE2, LE3,LP1, LP2, LP3          ;
LIST  LIST1 (K, A1, B1, C[K], OMEGA),
      LIST2(FOR I ← 0 STEP 1 UNTIL N[R]-1 DO GEN[I]),
      LIST3(FOR I ← 0 STEP 1 UNTIL N[R]-1 DO H[0,I]),
      LIST4(FOR I ← 0 STEP 1 UNTIL N[R]-1 DO H[1,I]),
      LIST5(FOR I ← 0 STEP 1 UNTIL M-1 DO [BLOCK[I],TIM[I]])                 ;
```

```
FORMAT OUT FMT1(/"REFERENCE GENERATOR FREQUENCY APPROXIMATION"),
           FMT2(10(F7.1,X4)),
           FMT3(/"ARRAY GEN[*] - GENERATOR OUTPUT"),
           FMT4(/"ARRAY H[0,*] - 30 KC PM TUBE"),
           FMT5(/"ARRAY H[1,*] - 40 KC PM TUBE"),
           FMT6(/"CALIBRATION TIMES AND BLOCK NUMBERS"),
           FMT7(5(I5,X2,F10.3,X4)),
           FMT8(I2, X2, 3(E15.8,X4),X3, F10.6),
           FMT9(/"START TIME TOO LARGE"),
           FMT10(/"END OF FILE -- R= ", I3),
           FMT11(/"END OF FILE - STAGE 2 -- R= ", I3),
           FMT12(/"PROGRAM ERROR - END OF FILE CALIBRATION TAPE"),
           FMT13(/"PARITY ERROR BEFORE STARTING TIME -- BLOCK: ", I4),
           FMT14(/"PARITY ERROR STAGE 1 - BLOCK: ",I4,X3,"R= ",I3),
           FMT15(/"PARITY ERROR STAGE 2 - BLOCK: ",I4,X3,"R= ",I3),
           FMT16(////"# ",I2," RELATIVE ANGLE OF INCIDENT POLARIZATION",
                 " -- ", F8.3, " RADIANS"),
           FMT17(//"NUMBER OF SAMPLES PER ANGLE OF INCIDENT",
                 " POLARIZATION: ",I4/"NUMBER OF ANGLES OF INCIDENT",
                 " POLARIZATION: ", I3)                                ;
      L ← 0                                                            ;
      R ← -1                                                           ;
      P ← (M1-3) DIV 4 - 1                                             ;
L1:   READ (FLDT1,2,A[*])[LE1:LP1]                                     ;
      IF A[1].[30:17] < STRTTM                         THEN
           GO TO L1                                                    ;
      SPACE (FLDT1,-1)[L5]                                             ;
L2:   READ (FLDT1[NO], 1, A[*])[LE2:LP2]                               ;
      WORD (FLDT1(0), M1)                                              ;
      READ (FLDT1, M1, A[*])                                           ;
      IF Z1 AND A[M1-1].[1:13] < LAMBDA[L]             THEN
           GO TO L2                                                    ;
      FOR J ← 3 STEP 4 UNTIL M1-1 DO                             BEGIN
           IF A[J+1].[14:11] > LEVEL                   THEN
                IF NOT Z1                              THEN  BEGIN
                     R ← R+1                                           ;
                     Z1 ← TRUE                               END       ;
```

```
            IF Z1                                                 THEN
            IF A[J].[1:13] = LAMBDA[L]                            THEN  BEGIN
                TIM[R] ← A[1].[30:17] + A[J].[25:8] - A[3].[25:8]
                       + A[J].[39:10]/1000                                  ;
                BLOCK[R] ← A[0]                                             ;
                IF R=0 AND L=0                                    THEN
                    FOR I ←0 STEP 1 UNTIL R DO
                        G[I] ← A[4×I+3].[14:11]                             ;
                K ← J-2                                                     ;
                FOR I ← 0 STEP 1 UNTIL N[R]-1 DO                        BEGIN
                    IF 4×I+K > M1-2                               THEN  BEGIN
                        READ (FLDT1, M1, A[*])[LE3:LP3]                     ;
                        SPACE(FLDT1,-1)                                     ;
                        K ← 5 - 4×I                                     END ;
L3:                 IF A[4×I+K].[43:5] ≠ 3                        THEN  BEGIN
                        K ← K + 1                                           ;
                        GO TO L3                                        END ;
                    GEN[I] ← A[4×I+K-2].[14:11]                             ;
                    H[0,I] ← A[4×I+K].[14:11]                               ;
                    H[1,I] ← A[4×I+K+1].[14:11]                             ;
                                                                        END ;
        IF Z[0]                                                   THEN  BEGIN
            WRITE (FIL2, FMT16, R, RSI[R])                                  ;
            WRITE (FIL2, FMT3)                                              ;
            WRITE (FIL2, FMT2, LIST2)                                       ;
            WRITE (FIL2, FMT4)                                              ;
            WRITE (FIL2, FMT2, LIST3)                                       ;
            WRITE (FIL2, FMT5)                                              ;
            WRITE (FIL2, FMT2, LIST4)                                   END ;
                WRITE (FTAPE1, N[R], GEN[*])                                ;
                WRITE (FTAPE1, N[R], H[0,*])                                ;
                WRITE (FTAPE1, N[R], H[1,*])                                ;
                SPACE (FLDT1,-1)                                            ;
                L ← L +1                                                    ;
                IF L < U                                          THEN
                    GO TO L2                                      ELSE  BEGIN
                    Z1 ← FALSE                                              ;
```

```
                     IF R < M-1                          THEN  BEGIN
                          L ← 0                                            ;
                          GO TO L2                             END
                                                         ELSE
                          GO TO L4                             END         ;
                                                               END   END   ;
      GO TO L2                                                             ;
LE1:  WRITE (FIL2, FMT9)                                                   ;
      SENS ← 1                                                             ;
      GO TO L4                                                             ;
LE2: WRITE (FIL2, FMT10, R)                                                ;
      IF R > 6                                           THEN
           M ← R + 1                                     ELSE
           SENS ← 1                                                        ;
      GO TO L4                                                             ;
LE3: WRITE (FIL2, FMT11, R)                                                ;
      IF R > 7                                           THEN
           M ← R                                         ELSE
           SENS ← 1                                                        ;
      GO TO L4                                                             ;
L5:   WRITE (FIL2, FMT12)                                                  ;
      SENS ← 1                                                             ;
      GO TO L4                                                             ;
LP1:  RELEASE (FLDT1)                                                      ;
      WRITE (FIL2, FMT13, A[0])                                            ;
      GO TO L1                                                             ;
LP2:  RELEASE (FLDT1)                                                      ;
      WRITE (FIL2, FMT14, A[0], R)                                         ;
      Z1 ← FALSE                                                           ;
      FOR I ← R STEP 1 UNTIL M-2 DO
           PSI[I] ← PSI[I+1]                                               ;
      M ← M-1                                                              ;
      IF M > 8                                           THEN  BEGIN
           SENS ← 1                                                        ;
           GO TO L4                                            END         ;
      IF R < M                                           THEN
           GO TO L2                                      ELSE
```

```
          GO TO L4                                                  ;
LP3:    RELEASE (FLDT1)                                             ;
        WRITE (FIL2, FMT15, A[0], R)                                ;
        Z1 ← FALSE                                                  ;
        FOR I ← R STEP 1 UNTIL M-2 DO
             PSI[I] ← PSI[I+1]                                      ;
        M ← M-1                                                     ;
        IF M > 8                                THEN  BEGIN
             SENS ← 1                                               ;
             GO TO L4                                 END           ;
        IF R < M                                THEN
             GO TO L2                           ELSE
             GO TO L4                                               ;
L4:     REWIND (FTAPE1)                                             ;
        IF Z[1]                                 THEN        BEGIN
             WRITE (FIL2, FMT17, SMP, M)                            ;
             WRITE (FIL2, FMT6)                                     ;
             WRITE (FIL2, FMT7, LIST5)                      END     ;
        LOCK (FLDT1, RELEASE)                                       ;
        OMIN ← 0.9×OMGA                                             ;
        OMAX ← 1.1×OMGA                                             ;
        S ← (OMAX-OMIN)/10.0                                        ;
        IF Z[2]                                 THEN
             WRITE (FIL2, FMT1)                                     ;
L6:     K ← 0                                                       ;
        FOR OMEGA ← OMIN STEP S UNTIL OMAX DO               BEGIN
             A1 ← B1 ← 0                                            ;
             FOR I ← 0 STEP 1 UNTIL R DO              BEGIN
                  A1 ← A1 + G[I]×COS(I×OMEGA)                       ;
                  B1 ← B1 + G[I]×SIN(I×OMEGA)         END           ;
             C[K] ← A1*2 + B1*2                                     ;
        IF Z[2]                                 THEN
             WRITE(FIL2, FMT8, LIST1)                               ;
             K ← K + 1                                      END     ;
             A1 ← C[0]                                              ;
             FOR I ← 1 STEP 1 UNTIL K-1 DO
                  IF C[I] > A1                  THEN        BEGIN
```

```
                   A1 ← C[I]                                                    ;
                   OMGA ← OMIN + I×S                                        END ;
     IF (OMAX-OMIN)/OMGA < 0.01                        THEN
          GO TO L7                                                              ;
     OMIN ← OMGA - S                                                            ;
     OMAX ← OMGA + S                                                            ;
     S ← S/10.0                                                                 ;
     GO TO L6                                                                   ;
L7:END *                                                                        ;
     IF SENS = 1                                       THEN
          GO TO L1                                                              ;
     FOR L ← 0 STEP 1 UNTIL U-1 DO
BEGIN INTEGER I,J,Q, K                                                          ;
     ARRAY GEN[0:SMP-1],PM[0:1,0:M-1,0:SMP-1], A[0:7],
          X[0:7,0:IF M>SMP THEN M-1 ELSE SMP-1],THETA2[0:M-1,0:SMP-1];
     REAL PHI, AMP, R, OMEGA                                                    ;
     INTEGER ARRAY B[0:SMP]                                                     ;
     LIST LIST1(FOR I ← 0 STEP 1 UNTIL N[J]-1 DO THETA2[J,I])                   ;
FORMAT OUT FMT1(/"REFERENCE GENERATOR DATA -- SMALL ANGLE APPROXIMATION"
              ),
          FMT2(9(F10.5,X2)),
          FMT3(/"REFERENCE GENERATOR CURVEFITTING DATA"),
          FMT4(3(F12.6,X4)/X2,3(F12.6,X2))                                      ;
     SPACE (FTAPE1, 3×L)                                                        ;
     FOR J ← 0 STEP 1 UNTIL M-1 DO                                         BEGIN
          READ (FTAPE1, N[J], GEN[*])                                           ;
          READ (FTAPE1, N[J], PM[0,J,*])                                        ;
          READ (FTAPE1, N[J], PM[1,J,*])                                        ;
          IF J ≠ M-1                                   THEN
               SPACE(FTAPE1, 3×(U-1))                                           ;
     FOR I ← 0 STEP 1 UNTIL SMP-1 DO                                       BEGIN
          X[0,I] ← 1.0                                                          ;
          X[1,I] ← SIN(I×OMGA)                                                  ;
          X[2,I] ← COS(I×OMGA)                                                  ;
          FOR K ← 1,2 DO                                              BEGIN
               X[2×K+1,I] ← X[1,I]×(I*K)                                        ;
               X[2×K+2,I] ← X[2,I]×(I*K)                              END       ;
```

```
      X[7,I] ← GEN[I]                                                    END  ;
COMPUTE (N[J],7,X,1.2,A,B,UPT[0],FIL2)                                        ;
PHI ← ATAN(A[2],A[1])                                                         ;
IF ABS(A[4]-A[3]) > 0.001                                   THEN
      IF ABS(A[2]) > ABS(A[1])                              THEN
            OMGA ← OMGA - A[3]/A[2]                         ELSE
            OMGA ← OMGA + A[4]/A[1]                                           ;
IF Z[3]                                                     THEN         BEGIN
      WRITE (FIL2, FMT1)                                                      ;
      WRITE (FIL2, FMT2, PHI,OMGA,FOR I←0,1,2,3,4,5,6 DO A[I])END ;
FOR I ← 0 STEP 1 UNTIL B[0]-1 DO                                         BEGIN
      X[1,I] ← SIN(OMGA×B[I+1])                                               ;
      X[2,I] ← COS(OMGA×B[I+1])                                               ;
      X[3,I] ← GEN[B[I+1]]                                               END  ;
COMPUTE (B[0],3,X,1.2,A,B,UPT[1],FIL2)                                        ;
PHI ← ATAN(A[2],A[1])                                                         ;
AMP ← SQRT(A[1]*2+A[2]*2)                                                     ;
IF Z[4]                                                     THEN         BEGIN
      WRITE (FIL2, FMT3)                                                      ;
      WRITE (FIL2, FMT4,A[0],AMP,PHI,FOR I←0,1,2 DO A[I])                END  ;
FOR I ← 0 STEP 1 UNTIL N[J]-1 DO                                         BEGIN
      R ← (GEN[I]-A[0])/AMP                                                   ;
      OMEGA ← PHI + I×OMGA                                                    ;
      IF I > 1                                              THEN
            OMEGA ← (2×THETA2[J,I-1]-THETA2[J,I-2]+OMEGA)/2                   ;
      IF ABS(R) ≤ 1.0                                       THEN  BEGIN
            R ← PI/2 - ARCCOS(R)                                              ;
            K ←ENTIER(OMEGA/PI2)                                              ;
            IF K ≤ 0                                        THEN
            THETA2[J,I] ← R                                 ELSE  BEGIN
                  K ← K+1                                                     ;
                  IF K MOD 4 ≤ 1                            THEN
                        THETA2[J,I] ← PI×(K DIV 2) + R
                                                            ELSE
                        THETA2[J,I] ← PI×(K DIV 2) - R
                                                                  END
                                                                  END
```

```
                                                                    ELSE
               THETA2[J,I] ← OMEGA                                               END  ;
          IF Z[4]                                                   THEN
               WRITE (FIL2, FMT2, LIST1)                                              ;
                                                                                 END  ;
     REWIND (FTAPE1)                                                                  ;
     IF L = U-1                                                     THEN         BEGIN
          BREAK                                                                       ;
          BKTIM ← TIME(2)                                                             ;
          CLOSE (FTAPE1, PURGE)                                                  END  ;
     FOR W ← 0 STEP 1 UNTIL 1 DO                                                 BEGIN
     IF TIME(2) - BKTIM > 3600×BRKTIM                               THEN         BEGIN
          BREAK                                                                       ;
          BKTIM ← TIME(2)                                                        END  ;
     IF LAMBDA[L] > LAMBDA2                                         THEN
          Q ← 1                                                                       ;
     IF Q = 0                                                       THEN
          TUBE ← "541F"                                             ELSE
          TUBE ← "541A"                                                               ;
     FOR J ← 0 STEP 1 UNTIL M-1 DO
          FOR I ← 0 STEP 1 UNTIL N[J]-1 DO
               PM[Q,J,I] ← CALCPM(Q,PM[Q,J,I])                                        ;
BEGIN INTEGER I,J,K                                                                   ;
      LABEL L1,L2                                                                     ;
      REAL PSI2, ETA, XI21, XI22, ZETA, BETA2, RR2, RR, RP2, RP,
           COSDLT, DELTA, K1, CI, WOL, V                                              ;
      ARRAY A1,B[0:M-1,0:4],D[0:9], C[0:4,0:2],X1[0:1,0:M-1],A[0:4]                   ;
      INTEGER ARRAY U1,U3[0:SMP], U2,U4[0:4,0:M]                                      ;
LIST  LIST1(FOR J ← 0 STEP 1 UNTIL M-1 DO FOR K← 0,1,2,3,4 DO B[J,K]),
      LIST2(FOR I ← 0,1,2 DO FOR K ← 0,1,2,3,4 DO C[K,I]),
      LIST3(ETA, XI21, XI22, ZETA),
      LIST4(FOR J ← 0 STEP 1 UNTIL M-1 DO FOR K ← 0,1,2,3,4 DO A1[J,K]),
      LIST5(FOR I ← 0 STEP 1 UNTIL 9 DO D[I]),
      LIST6(FOR I←0 STEP 1 UNTIL 9 DO [I,D[I]], RR2, RP2),
      LIST7(FOR I←0 STEP 1 UNTIL 9 DO [I,D[I]], RR, COSDLT, RP2),
      LIST8(FOR I←0 STEP 1 UNTIL 9 DO [I,D[I]], RR, DELTA, RP2),
      LIST9 (LAMBDA[L], CI, RR, RP, DELTA, ZETA, ETA),
```

```
      LIST10(Q,LAMBDA[L], CI, RR, RP, DELTA, ZETA, ETA),
      LIST11(POL,V,PHI)                                                  ;
  FORMAT OUT FMT1(/"ARRAY B"),
             FMT2(5(E12.5, X4)),
             FMT3(/"ARRAY C"),
             FMT4("ETA ", F9.6,X4,"X12 ", 2(F8.6,X3)/"ZETA",F9.6/),
             FMT5(/"ARRAY A"),
             FMT6(/"ARRAY D"),
                  FMT7(/"COMPUTE ERROR - ARRAY D"/5(I2,X1,E12.5,X3)/
                       4(I2,X1,E12.5,X3)/I2,X1,E12.5,X3),
                  FMT8(/"COMPUTE ERROR - ARRAY D"/2(5(I2,X1,E12.5,X3)/),
                       "RR2= ", E12.5/"RP2= ", E12.5),
                  FMT9(/"COMPUTE ERROR - ARRAY D"/2(5(I2,X1,E12.5,X3)/),
                       "RR= ",F10.5/"COS(DELTA)= ",E12.5/"RP2= ",E12.5),
                  FMT10(/"COMPUTE ERROR - ARRAY D"/2(5(I2,X1,E12.5,X3)/),
                       "RR= ",F10.5/"DELTA= ",F10.6/"RP2= ", E12.5),
             FMT11(X4,A4),
             FMT12("LAMBDA: ", I5/"CI ",E13.6,X3,"RR ",F10.8,X3,"RP ",
                   F10.8,X3,"DELTA ", F12.8/X19,"ZETA ",F10.7,X1,"ETA ",
                   F10.7),
             FMT13(I2,X2,I4,X2,E13.6, 5(F10.7,X1)),
             FMT14(X63, "P ", F6.3, X4,"V ",F6.3," PHI",F7.3)            ;
      FOR I ← 0 STEP 1 UNTIL IF SMP>M THEN SMP-1 ELSE M-1 DO
             X[0,I] ← 1.0                                                ;
      IF Z[7+Q]                                          THEN      BEGIN
             ETA ← ETAAR[Q]                                              ;
             ZETA ←ZETAAR[Q]                                             ;
             U1[0] ← SMP                                                 ;
             FOR I ← 0 STEP 1 UNTIL SMP-1 DO
                 U1[I+1] ← I                                             ;
             FOR I ← 0,1,2,3,4 DO                            BEGIN
                 U2[I,0] ← M                                             ;
                 FOR J ←0 STEP 1 UNTIL M-1 DO
                     U2[I,J+1] ← J                           END         ;
             GO TO L1                                                END ;
      FOR J ← 0 STEP 1 UNTIL M-1 DO                                  BEGIN
             FOR I ← 0 STEP 1 UNTIL N[J]-1 DO                BEGIN
```

```
                X[1,I] ← SIN(THETA2[J,I])                                              ;
                X[2,I] ← COS(THETA2[J,I])                                              ;
                X[3,I] ← SIN(2×THETA2[J,I])                                            ;
                X[4,I] ← COS(2×THETA2[J,I])                                            ;
                X[5,I] ← PM[Q,J,I]                                     END             ;
            COMPUTE (N[J],5,X,0.9,B[J,*], U1,OPT[2],FIL2)                    END       ;
        WRITE (FIL2, FMT1)                                                             ;
        WRITE (FIL2, FMT2, LIST1)                                                      ;
        FOR J ← 0 STEP 1 UNTIL M-1 DO                                        BEGIN
            PSI2 ← 2×PSI[J]                                                            ;
            X[1,J] ← X1[0,J] ← SIN(PSI2)                                               ;
            X[2,J] ← X1[1,J] ← COS(PSI2)                                     END       ;
        FOR I ← 0,1,2 DO                                                     BEGIN
            FOR J ← 0 STEP 1 UNTIL M-1 DO
                X[3,J] ← B[J,I]                                                        ;
            COMPUTE (M,3,X,0.9,C[I,*],U2[I,*],OPT[3],FIL2)                   END       ;
        FOR I ← 3,4 DO                                                       BEGIN
            FOR J ← 0 STEP 1 UNTIL M-1 DO                              BEGIN
                X[0,J] ← X1[0,J]                                                       ;
                X[1,J] ← X1[1,J]                                                       ;
                X[2,J] ← B[J,I]                                        END             ;
            COMPUTE (M,2,X,0.9,C[I,*],U2[I,*],OPT[3],FIL2)                   END       ;
        WRITE (FIL2, FMT3)                                                             ;
        WRITE (FIL2, FMT2, LIST2)                                                      ;
        ETA ← 0.5×ATAN(-C[0,1],C[0,2])                                                 ;
        XI21 ← 0.5×ATAN(C[3,1],C[3,0])                                                 ;
        XI22 ← 0.5×ATAN(-C[4,0],C[4,1])                                                ;
        IF ABS(XI21-XI22) < 0.26                                   THEN
            ZETA ← (ETA - (XI21+XI22)/2)/2                         ELSE
            IF ABS(C[3,1]) > ABS(C[4,2])                           THEN
                ZETA ← (ETA - XI21)/2                              ELSE
                ZETA ← (ETA - XI22)/2                                                  ;
        WRITE (FIL2, FMT4, LIST3)                                                      ;
        FOR I ← 0 STEP 1 UNTIL IF SMP>M THEN SMP-1 ELSE M-1 DO
            X[0,I] ← 1.0                                                               ;
L1:     FOR J ← 0 STEP 1 UNTIL M-1 DO                                        BEGIN
            FOR I ← 0 STEP 1 UNTIL W1[0]-1 DO                          BEGIN
```

```
          BETA2 ← ZETA×2 + THETA2[J,U1[I+1]]                              ;
          X[1,I] ← SIN(BETA2)                                             ;
          X[2,I] ← COS(BETA2)                                             ;
          X[3,I] ← SIN(2×BETA2)                                           ;
          X[4,I] ←COS(2×BETA2)                                            ;
          X[5,I] ← PM[Q,J,U1[I+1]]                          END           ;
     COMPUTE(U1[0],5,X,0.9,A1[J,*],U3,OPT[4], FIL2)               END     ;
WRITE (FIL2, FMT5)                                                        ;
WRITE (FIL2, FMT2, LIST4)                                                 ;
FOR I ← 0 STEP 1 UNTIL M-1 DO                                     BEGIN
     X1[0,I] ← SIN(2×(PSI[I] + ETA))                                      ;
     X1[1,I] ← COS(2×(PSI[I] + ETA))                              END     ;
FOR I ← 0,1,2 DO                                                  BEGIN
     FOR J ←0 STEP 1 UNTIL U2[I,0]-1 DO                     BEGIN
          X[1,J] ← X1[IF I=1 THEN 0 ELSE 1,U2[I,J+1]]                     ;
          X[2,J] ← A1[U2[I,J+1],I]                          END           ;
     COMPUTE (U2[I,0],2,X,0.9,A,U4[I,*],OPT[5],FIL2)                      ;
     FOR J ← 0,1 DO
          D[2×I+J] ← A[J]                                         END     ;
FOR I ← 3,4 DO                                                    BEGIN
     FOR J ← 0 STEP 1 UNTIL U2[I,0]-1 DO                    BEGIN
          X[0,J] ← X1[IF I=4 THEN 1 ELSE 0,U2[I,J+1]]                     ;
          X[1,J] ← A1[U2[I,J+1],I]                          END           ;
     COMPUTE (U2[I,0],1,X,0.9,A,U4[I,*],OPT[5],FIL2)                      ;
     D[2×I] ←A[0]                                                 END     ;
WRITE (FIL2, FMT6)                                                        ;
WRITE (FIL2, FMT2, LIST5)                                                 ;
FOR I ←0 STEP 1 UNTIL 9 DO
     D[I] ← ABS(D[I])                                                     ;
D[3] ← (D[3] + D[5])/2                                                    ;
D[5] ← (D[6] + D[8])/2                                                    ;
D[6] ← (D[0] - D[4])/(D[0] + D[4])                                        ;
D[7] ← (D[1] + D[5] - D[3])/(D[1] + D[5] + D[3])                          ;
D[8] ← D[6]×D[7]                                                          ;
D[9] ← (1 - D[6]*2)×(1 - D[7]*2)                                          ;
IF D[9] ≥ 0                                            THEN
     D[9] ← SQRT(D[9])                                 ELSE       BEGIN
```

```
          WRITE (FIL2, FMT7,LIST5)                                          ;
          GO TO L2                                                  END   ;
     WRITE (FIL2, FMT6)                                                     ;
     WRITE (FIL2, FMT2, LIST5)                                              ;
     RP2 ← (1 - D[8] - D[9])/(D[6] - D[7])                                 ;
     RR2 ← (1 + D[8] - D[9])/(D[6] + D[7])                                 ;
     IF RR2 ≥ 0                                        THEN
          RR ← SQRT(RR2)                               ELSE         BEGIN
          WRITE (FIL2, FMT8, LIST6)                                         ;
          GO TO L2                                                  END   ;
     COSDLT ← ((D[1]-D[5])/(D[1]+D[5]))×(1+RR2)/(2×RR)                     ;
     IF ABS(COSDLT) ≤ 1.0                              THEN
          DELTA ← ARCCOS(COSDLT)                       ELSE         BEGIN
          WRITE (FIL2, FMT9, LIST7)                                         ;
          GO TO L2                                                  END   ;
     IF RP2 ≥ 0                                        THEN
          RP ← SQRT(RP2)                               ELSE         BEGIN
          WRITE (FIL2, FMT10, LIST8)                                        ;
          GO TO L2                                                  END   ;
     CI ← D[0]/((1 + RR2)×(1 + RP2))                                        ;
     WRITE (FIL2, FMT11, TUBE)                                              ;
     WRITE (FIL2, FMT12, LIST9)                                             ;
     IF Z[5]                                           THEN
          WRITE (FIL4, FMT13, LIST10)                                       ;
     FOR J ← 0 STEP 1 UNTIL M-1 DO                                  BEGIN
          PHI ← ATAN(A1[J,3],A1[J,4])                                       ;
          IF PHI < 0                                   THEN
               PHI ← PHI + 6.28318534                                       ;
          POL ← 2×SQRT(A1[J,3]*2+A1[J,4]*2)/(CI×(1-RP2)×(1+RR2-2×RR×
                    COSDLT))                                                ;
          V ← (A1[J,1]-CI×(1+RP2)×(1-RR2)×POL×SIN(PHI))/(2×CI×RR×
                    (1-RP2)×SIN(DELTA))                                     ;
     PHI ← 0.5×PHI                                                          ;
          WRITE (FIL2, FMT14, LIST11)                               END   ;
L2:END                                                                      ;
     IF LAMBDA[L] < LAMBDA1                            THEN
          Q ← 1                                                             ;
```

```
                                                                      END  ;
END                                                                        ;
      IF Z[5]                                                 THEN     BEGIN
           REWIND (FIL4)                                                   ;
           IF Z[6]                                            THEN  BEGIN
                WRITE (FIL2[PAGE])                                         ;
                DATE (FIL2)                                                ;
                DAYTIM (FIL2)                                              ;
                WRITE (FIL2[DBL], FMT3)                             END    ;
BEGIN LABEL L1,L2                                                          ;
      FILE FIL6 0 (3,10)                                                   ;
      ARRAY A[0:9]                                                         ;
      DATE (FIL6)                                                          ;
      DAYTIM (FIL6)                                                        ;
      WRITE (FIL6, FMT3)                                                   ;
L1:   READ (FIL4, 10, A[*])[L2]                                            ;
      WRITE (FIL6, 10, A[*])                                               ;
      IF Z[6]                                                 THEN
           WRITE (FIL2, 10, A[*])                                          ;
      GO TO L1                                                             ;
L2:   IF Z[6]                                                 THEN
           WRITE (FIL2[PAGE])                                              ;
      CLOSE (FIL4, PURGE)                                                  ;
END                                                                        ;
                                                                      END  ;
L1:END                                                                     ;
      RNTIM2 ← TIME(2)/60                                                  ;
      WRITE (FIL2, FMT1, RNTIM2)                                           ;
      RNTIM3 ←TIME(3)/60                                                   ;
      WRITE (FIL2, FMT2, RNTIM3)                                           ;
END .
```

BIBLIOGRAPHY

1. W. G. Fastie and G. H. Dieke, Final Report, Contract AF19(604)-5516, AFCRL 1060 (31 July 1961)

2. W. G. Fastie, Journal of the Optical Society of America, 42, 9, 641-647 (Sept. 1952)

3. Z. Sekera, Final Report, Contract AF19(122)-239, Dept. of Meteorology, University of California (1955)

4. C. R. N. Rao and Z. Sekera, Scientific Report No. 1, Contract AF19(604)-8050, AFCRL 63 809 (June 1963)

5. Z. Sekera, C. R. N. Rao and D. Dribble, Review of Scientific Instruments, 35, 764 (1963)

6. E. B. Hodgdon and H. D. Edwards, Final Report, Contract AF19(628)-2416, AFCRL 66 397 (April 1966)

7. R. Stain, W. E. Schneider and J. K. Jackson, Applied Optics, 2, 1151 (1963)

8. P. Bener, Tech. Note 2, Contract AF61(052)-618, AFCRL 63 654, Davos-Plate, Switzerland (Jan. 1963)

9. C. G. Stokes, Transactions of the Cambridge Philosophical Society, 9, 399 (1852)

10. H. Mueller, Journal of the Optical Society of America, 38, 661 (1948)